Semiannual Status Report
for
NAG 5–1612

Submitted to:
Instrument Division
Engineering Directorate
Goddard Space Flight Center
Greenbelt, MD 20771
Attn: Warner Miller and Pen Shu Yeh

K. Sayood, Y.C. Chen and G. Kipp

Department of Electrical Engineering
and
Center for Communication & Information Science
University of Nebraska-Lincoln
Lincoln, Nebraska 68588–0511

Period: June 15, 1992 — December 15, 1992

1 Introduction

During this period we began the devolopment of simulators for the various HDTV systems proposed to the FCC. This simulators will be tested using test sequences from the MPEG committee. The results will be extrapolated to HDTV video sequences.

Currently we have completed the simulator for the compression aspects of the *Advanced Digital Televsion* (ADTV) proposed by David Sarnoff Laboratories, NBC, North American Phillips, and Thomson Consumer Electronics. This proposal has also been supported by Texas Instruments. We are at various stages of development in terms of the other HDTV proposals.

In the following section we give a brief overview of the ADTV system. In section 3, we look at some coding results obtained using the simulator. We compare these results to those obtained using the CCITT H.261 standard. In section 4, we evaluate these results in the context of the CCSDS specifications and make some suggestions as to how the ADTV system could be implemented in the NASA network.

Some caveats are in order here. The sequences we used for testing the simulator and generating the results are those used for testing the MPEG algorithm. The sequences are of much lower resolution than the HDTV sequences would be, and therefore the extrapolations are not totally accurate. We would expect to get significantly higher comperession in terms of bits per pixel with sequences that are of higher resolution. However, the simulator itself is a valid one, and should HDTV sequences become available, they could be used directly with the simulator.

In addition to the work described in this report the following papers appeared or were submitted during the past six months, describing other work under this grant and its predecessor. The papers are included in the appendix.

- Y.C. Chen, K. Sayood, and D.J. Nelson, "A Robust Coding Scheme for Packet Video",*IEEE Transactions on Communications*, vol. 40, pp. 1491-1501, September 1992.
- K. Sayood, "Data Compression in Remote Sensing Applications," *IEEE Geoscience and Remote Sensing Society Newsletter*, no. 84, pp. 7-15, September 1992.

- A.C. Hadenfeldt and K. Sayood, "Compression of Color-Mapped Images," submitted to *IEEE Transactions on Geoscience and Remote Sensing*

- K. Sayood and S. Na, "Recursively Indexed Differential Pulse Code Modulation," *Proceedings IEEE/DIMACS Workshop on Quantization and Coding*, Piscataway NJ, November 1992.

- K. Sayood, F. Liu, and J.D. Gibson, "A Joint Source Channel Coder Design," accepted for 1993 International Conference on Communication, Geneva, Switzerland, May 1993.

- A.C. Hadenfeldt and K. Sayood, "Compresion of Color-Mapped Images," accepted for *1993 International Conference on Communication*, Geneva, Switzerland, May 1993.

- B. Gorjala, K. Sayood, and G. Meempat, "An Image Compression Scheme for use in Token Ring Networks", submitted to *1993 International Conference on Communication*, Geneva, Switzerland, May 1993.

2 Advanced Digital Television

There are three key elements in the ADTV system.

- ADTV uses MPEG++(Moving Pictures Expert Group) draft proposal as its compression scheme.

- ADTV incorporates a Prioritized Data Transport(PDT) which is a cell relay-based data transport layer to supports the prioritized delivery of video data. PDT also offers service flexibility and compatibility to broadband ISDN.

- ADTV applies spectral-shaping techniques to Quadrature Amplitude Modulation(QAM) to minimize interference from and to any co-channel NTSC signals.

We have simulated all aspects of the compression algorithm of the ADTV proposal. The compression algorithm as described in the *Advanced Digital*

Television, System Description submitted to FCC/ACATS and as implemented in the simulator is described below.

2.1 Compression Algorithm

The basic compression approach is the MPEG++ algorithm which upgrades the standard MPEG approach to HDTV performance level. The key components of this algorithm are described below.

2.1.1 Group of Pictures(GOP)

A GOP comprises up to three types of frames, the I, P and B frames. The I frames are processed using only intra-frame DCT coder with adaptive quantization; the P frames are processed using a hybrid temporal predictive DCT coder with adaptive quantization and forward motion compensation; the B frames are processed using a hybrid temporal predictive DCT coder with adaptive quantization and bidirectional motion compensation. The I and P frames are referred to as the anchor frames because of their roles in the bidirectional motion compensation of the B frames. The GOP structure offers a good tradeoff between the high efficiency of temporal predictive coding, good error-concealment features of periodic intra-only processing, and fast picture acquisition.

2.1.2 Input Sequencer

The GOP data structure requires some unique sequencing of the input video frames. Because of the backward motion compensation in B frames processing, the anchor frames must be processed before the B frames associated with the two anchors. The frames are transmitted in the same order as they are processed.

2.1.3 Raster Line to Block/Macroblock Converter

The basic DCT transform unit is an 8 x 8 pixel block called a block. The basic quantization unit is four adjacent blocks of Y, and one U and one

V blocks. Such a quantization unit is called a macroblock. The converter converts the raster line format to the block and the macroblock format.

2.1.4 I-frame Processing

An I frame is processed by an intra-frame DCT coder without motion compensation. A fixed quantizer is applied to the DC coefficient. The AC values are first weighted by a down- loadable quantization matrix before uniform adaptive quantization. The quantization step for the AC coefficients is controlled by a Rate Controller. The I frame coding is pretty much the same as JPEG scheme.

2.1.5 P-frame Processing

A P frame is first processed by forward motion compensation, motion is always referenced to the nearest past anchor frame. The search area is proportional to the number of B frames between two consecutive anchor frames. The prediction residue or original macroblock, depending on the motion compensation result, goes to DCT coder and quantizer. For intra-macroblocks, the DCT coefficient quantization is identical to that used for the I frames. For motion-compensated macroblocks, the DC and AC coefficients are quantized with same uniform quantizer.

2.1.6 B-frame Processing

Unlike the P frames, the B frames are subjected to bidirectional motion compensation. The motion references are the two anchor frames sandwiching the B frames. The search regions are proportional to the temporal distance between the B frame and the two anchor frames. Like P-frame macroblock, the B-frame macroblocks have a number of modes. In addition to all the modes for a P-frame macroblock, the B-frame macroblock further includes a bidirectional interpolative mode, using both forward and backward motion compensation, and a unidirectional mode. In the interpolative mode, an average of the forward and the backward motion- compensated macroblocks is used as the prediction macroblock. The B-frame macroblock is processed as a P-frame macroblock.

2.1.7 Differential, Run-Length and Variable-Length Coding(VLC)

The quantized DC coefficients of all the I-frame macroblock and P-, B-frame macroblocks in intra mode are coded with a DPCM coder. The quantized AC coefficients are coded with VLC after the zigzag scan ordering. Motion vectors are differentially coded. In addition, VLC is applied to all the coded information: motion vectors, macroblock addresses, block types, etc.

2.2 Data Prioritization Layer

The Prioritization Layer comprises the Priority Processor and the Rate Controller.

2.2.1 Priority Processor

Based on the information from the Rate Controller, the Priority processor pre-calculates the rate of HP(High Priority)/LP(Low Priority) for every frame. HP/LP fractional allocations may vary with the frame type. Every data element gets a priority assignment from the Priority Processor according to its importance. The header is always most important, followed by the motion vector, DC value, low frequency coefficients and high frequency coefficients.

2.2.2 Rate Controller

The Rate Controller monitors the status of the rate buffers in the Transport Encoder. It uses the buffer occupancy information to compute the necessary compression requirement and feeds the results in the form of appropriate quantization parameters to the Video Processor in the Compression Encoder. The Rate Controller also provides input to the Priority Processor regarding the initial allocation of HP/LP rate for the next Group of Pictures.

The algorithm used in this simulation for rate control is given by

$$QS = 2\lceil \frac{B}{200p} \rceil + 2$$

where QS is the quantizer step-size, and B is a measure of buffer fullness.

2.3 Transport Layer

The Transport Layer comprises the Transport Processor and the Rate Buffer.

2.3.1 Transport Processor

Data elements are supplied to the Transport Processor from the Prioritization Processor. The Transport Processor generates appropriate header fields for data group. The header fields are used in the construction of a basic transport unit called cell. A cell has a header and a trailer enclosing a payload area. Each cell has a fixed size of 256 bytes long. The header contains chaining and segmentation information which allows data groups to be segmented across cells. This feature limits the propagation of channel error from one cell to the next. The trailer field contains 16-bit error-checking CRC code.

2.3.2 Rate Buffer

Since the number of cells generated is not constant and the channel coding module interfaces with the Transport Processor at a fixed clock rate, we need a buffer to smooth out any rate variation. The maximum end-to-end delay is dependent on the size of the buffer.

3 Simulation Results

The ADTV system described above without the priority and transport processors was simulated in detail. The simulation programs were written in C and implemented on a *SUN* workstation. Along with the ADTV system we also simulated a video coding scheme based on the CCITT H.261 recommendations. The purpose was to have a benchmark for simulation.

In our ADTV simulator, the frames were arranged in the following sequence

I B B P B B P B B P B B I B B P...

The sequence used for testing the simulator was the *Susie* sequence. This sequence contains both low and moderate motion of the type to be encountered in most NASA applications. We present the results in the form of graphs tables and a videotape accompanying this report.

The coding rates and PSNR under different coding conditions are listed in Table 1. Two parameters are used to control the coding conditions. The parameter p controls the output rate and length of the rate buffer. The fullness of the rate buffer determines the quantizer step-size and therefore, the coding rate and quality. Thus the parameter p has an important impact on both coding rate and quality. The parameter t is used to decide whether the macroblock after motion compensation needs coding. Smaller values of t will lead to higher rates, while smaller values of t will result in lower quality.

The first three sequences were coded using the ADTV algorithm. In Sequence 1, with p=3, the average rate is is 0.22 bits/pixel. This rate is not sufficient to effectively code the I frames. As the B and P frames depend heavily on the I frames, this has a cascade effect on the entire sequence. As the quantizer stepsize depends on how full the buffer is, this low rate leads to the buffer getting filled up as the lower portions of the I frame are being coded. This means that when coding the lower regions of the I frames, the quantizer is coarse. This results in blocking artifacts which are very noticeable in the lower portions of the sequence. This effect can be seen in the first sequence on the videotape. When the coder has finished with the I frame, and the B and P frames are being coded, as the coding rate is lower for the B and P frames, the buffer situation gets partly remedied. However, this is not sufficient to get the quantizer step-size small enough to remove the blocking effect. When the buffer size is increased (p=5) the number of artifacts is reduced as seen in the second sequence on the video tape. An interesting effect can be seen in the third sequence. Here the buffer size was kept the same as in the second sequence, however, the motion compensation threshold t was kept high. This means that blocks that would have been coded in sequence 2 are left uncoded in sequence 3. This in turn emphasizes the blocking effects. One would think that given the fact that we are accepting more distortion, the rate would go down. However as we can see from Figure 1, the rates for sequence 2 and sequence 3 are almost identical. This could be attributable to the fact that the poor reconstruction of the P frames lead to poor prediction and hence an increase in bit rate which takes away any savings from the higher motion

compensation threshold. Thus the parameter t while it effects the quality has little effect on the rate.

From Figure 1 we can see that the ADTV algorithm generates a very bursty traffic. This is in sharp contrast to the CCITT H.261 algorithm which produces relatively smooth output. We can see this from the results for sequences 4-7 which were coded using the CCITT H.261 algorithm using the same parameters p and t as the first three sequences. The rate and PSNR results for these sequences are given in Table 1 and Figures 3 and 4. Recall that the only significant difference between the CCITT H.261 algorithm and the ADTV algorithm are the sequencing and motion compensation. The use of the intra-frame coding every 13^{th} frame in the ADTV algorithm increases the bit rate. This is compensated for by using the different motion compensation approach giving the bursty traffic. In the CCITT H.261 algorithm, intra-frame coding is recommended once in every 134 frames, thus there is no significant variation in the rate from frame to frame. The disadvantage of the CCITT H.261 algorithm when compared to the ADTV algorithm is the decrease in the ability to randomly access any particular frame, and the decrease in the ability to react fast to sudden scene changes. Furhtermore it should be noted that the ADTV algorithm was proposed for HDTV sequences, while the sequences we are using have significantly less resolution.

Due to the importance of I frame, which serves as the anchor frame for both P and B frame, we decide to put more coding efforts in such frame to try to eliminate the blocking effect. In Sequence 7-9, the ADTV algorithm has been modified to keep the quantization stepsize QS constant while coding the I frame. One effect is that the buffer becomes really full during coding the I frame, and the subsequent frame gets very little of the coding resources. This results in an increase in burstiness as can be seen from Figures 5 and 6. However, this approach does result in the reduction/elimination of the blocking effect. That such a simple strategy can result in such dramatic improvement shows that should blocking effects appear in the HDTV sequences, attention should be paid to the encding of the I frames.

Figures 7-14 show various comparison results between the ADTV, The modified ADTV and the CCITT H.261 algorithms. While these comparisons show an advantage for the CCITT H.261 algorithm, subjective comparisons tend to show the reverse. We invite the reader to examine the videotape and

draw their own conclusions.

4 HDTV and CCSDS

To speculate how the HDTV service would be accomodated by the NASA network, we briefly review some of the relevant features of the CCSDS recommendations.

4.1 CCSDS Principal Network

A "CCSDS Principal Network" (CPN) serves as the project data handling network which provides end-to-end data flow in support of the Experimental, Observational and interactive users of Advanced Orbiting Systems. A CPN consists of an "Onboard Network" in an orbiting segment connected through a CCSDS "Space Link Subnetwork" (SLS) either to a "Ground Network" or to another Onboard Network in another orbiting segment. The SLS is the central component of a CPN; it is unique to the space mission environment and provides customized services and data communications protocols. Within the SLS, CCSDS defines a full protocol to achieve "cross support" between agencies. Cross support is defined as the capability for one space agency to bidirectionally transfer another agency's data between ground and space systems using its own transmission resources. A key feature of this protocol is the concept of a "Virtual Channel" which allows one physical space channel to be shared among several data streams, each of them may have different service requirements. A single Physical space channel may therefore be divided into several logical data channels, each known as a Virtual Channel.

Eight separate services are provided within a CPN. Two of these services ("Path" and "Internet") operate end-to-end across the entire CPN. They are complementary services, which satisfy different user data communications requirements: some users will interface with only one of them, but many will operate with both. The remaining six services ("Encapsulation", "Multiplexing", "Bitstream", "Insert", "Virtual Channel Data Unit" and "Physical Channel") are provided only within the Space Link Subnetwork for special

applications such as audio, video, high rate payloads, tape playback, and the intermediate transfer of Path and Internet data. Our interest is with the services provided by the Space Link Subnetwork.

4.2 Space Link Subnet Services

The Space Link Subnet supports the bidirectional transmission of data through the space/ground and space/space channels which interconnect the distributed elements of the CCSDS Advanced Orbiting Systems. It also provides "direct connect" transmission services for certain types of data which requires timely or high-rate access to the space channel. During SLS transfer, different flows of data are separated into different Virtual Channels, based on data handling requirements at the destination. These Virtual Channels are interleaved onto the physical channel as a serial symbol stream. A particular Virtual Channel may contain either packetized or bitstream data, or a combination of both.

The Space link Subnetwork consists of two layers; the Space Link layer and the Space Channel layer which correspond to ISO-equivalent Data Link layer and Physical layer respectively. Efficient use of the physical space channel was a primary driver in the development of these protocols. The Space Link layer is composed of the Virtual Channel Link Control sublayer(VCLC) and the Virtual Channel Access sublayer(VCA).

The main function of VCLC sublayer is to convert incoming data into a protocol data unit which is suitable for transmission over the physical space channel. Four type of protocol data units may be generated by the VCLC sublayer: fixed length blocks of CCSDS Packets, called "Multiplexing Protocol Data Units" (M-PDUs); fixed length blocks of Bitstream data, called "Bitstream Protocol Data Units" (B-PDUs); fixed length blocks of mixed packetized and isochronous data, called "Insert Protocol Data Units" (IN-PDUs); and fixed length blocks of data for use by retransmission control procedures, called "Space Link ARQ Procedure Protocol Data Units" (SLAP-PDUs).

There are several procedures in VCLC sublayer to perform the function. The Encapsulation procedure provides the flexibility to handle virtually any packet structure. It puts a primary header to delimited data units (including Internet packet) and make it to be a CCSDS Packet. Multiplexing procedure

multiplexes those CCSDS Packets on the same virtual channel together. The length of multiplexing protocol data unit is fixed since it is required to fit exactly in the fixed length data space of VCDU/CVCDU. There maybe some packets which overlap two or more M-PDU and "first packet pointer" points out where the first packet starts. Some user data, such as audio, video, playback and encrypted information, will simply be presented to the SLS as a stream of bits or octets. The Bitstream procedure simply blocks these data into individual Virtual Channels and transmits it. When the transmission rate is high, Bitstream data may be transmitted over a dedicated Virtual Channel. Alternatively, if the transmission rate is low, it can be inserted at the front of other packetized or bitstream data. This is called the Insert procedure. Through this procedure, bandwidth can be used more efficiently.

The last procedure of VCLC is Space Link ARQ Procedure (SLAP) which is used to provide guaranteed Grade-1 delivery of data links that interconnect the space and ground elements of a CPN. The SLAP-PDU carries "Link ARQ Control Words" (LACWs) which report progress on receipt of data flowing in the opposite direction. Upon arrival at the receiving end, the LACW is extracted from the PDU, and the sequence number is checked to assure that no data has been lost or duplicated. In the event of a sequence error, the LACW carried by PDUs traveling in the opposite direction is used to signal that a retransmission is required. This retransmission begins with the first PDU that was not received in sequence, and all subsequent PDUs are retransmitted in the order in which they were originally provided to the LSAP from the layer above.

The VCA sublayer creates the protocol data units used for space link data transfer: these are either "Virtual Channel Data Units" (VCDUs) or "Coded Virtual Channel Data Units" (CVCDUs), and are formed by appending fixed length Header, Trailer and (for CVCDUs) error correction fields to the fixed length data units generated by the VCLC sublayer. The VCA sublayer is composed of Virtual Channel Access(VCA) and Physical Channel Access(PCA) procedures. VCA procedure generates VCDU for protocol data units which come from VCLC sublayer or accepts independently generated VCDU from reliable users. A VCDU with a powerful outer code of error-correcting Reed-Solomon check symbols appended to it is called a CVCDU. Relative to a VCDU, a CVCDU contains more error-control information and, hence, less user data. "Virtual Channel ID" which is field of VCDU/CVCDU

Header can enable up to 64 VC to be run concurrently for each assigned Spacecraft ID on a particular physical space channel. Since space data is transmitted through weak signal, noisy channel as serial symbol stream, a robust frame synchronization process at the receiving end is required. Therefore, fixed length VCDU/CVCDU is used and PCA procedure prefixes a 32 bits Synchronization Marker in front of VCDU/CVCDU to form a "Channel Access Data Unit" (CADU). A contiguous and continuous stream of fixed length CADUs, known as a "Physical Channel Access Protocol Data Unit" (PCA-PDU) is transmitted as individual channel symbols through the ISO-equivalent Physical Layer of the Space Link Subnet, which is known as the "Space Channel Layer".

4.3 Space Link "Grades of Service"

Three different "grades of services" are provided by the Space Link Subnet, using a combination of error detection, error correction and retransmission control techniques. We have to note that each virtual channel can only support a single grade of service.

4.3.1 Grade-3 Service

This service provides the lowest quality of service. Data transmitted using Grade-3 service may be incomplete and there is a moderate probability that errors induced by the Space Link Subnet are present and that the sequence of data units is not preserved. A VCUD is discarded if an uncorrectable error is detected at the destination. Grade-3 service should not be used for transmission of asynchronous packetized data, because it provides insufficient protection for the extensive control information contained in the packet headers.

4.3.2 Grade-2 Service

CVCDU is the unit of transmission that support Grade-2 service. The Reed-Solomon encoding provides extremely powerful error correction capabilities. Data transmitted using Grade-2 service may be incomplete, but data se-

quencing is preserved and there is a very high probability that no data errors have been induced by the Space Link Subnet.

4.3.3 Grade-1 Service

Data transmitted using Grade-1 service are delivered through the Space Link Subnet complete, in sequence, without duplication and with a very high probability of containing no errors. It is provided by using two paired Reed-Solomon encoded Virtual Channels, in opposite directions, so that an Automatic Repeat Queueing (ARQ) retransmission scheme may be implemented.

5 HDTV Transmission on the CCSDS Network

As described in the previous section, some user data, such as audio, video, playback and encrypted information, can simply be presented to the Space Link Subnet (SLS) as a stream of bits or octets. The SLS merely blocks these data into individual Virtual Channels and transmits them using Bitstream Service. Some bitstream data, such as digitized video and audio, will have stringent delivery timing requirement and are known as "isochronous" data. For the transmission of ADTV coded information, the channel transmission rate is high enough to dedicate a specific Virtual Channel. Although the coding output rate is quite bursty, there are two mitigating circumstances

1. the pattern of burstiness is relativey "uniform". That is, the data rate peaks every 13^{th} frame.
2. the variations occur very fast, that is high traffic persists for only a single frame followed by low traffic.

Because of (2) the traffic can be smoothed out using a moderate sized buffer, and (1) implies that the size of the buffer can be ascertained with some confidence.

The delay constraints on the transmission preclude the use of the Space Link ARQ procedure, while the delay constraint coupled with the high rate argue

against the use of the Insert and Multiplexing procedures. This leaves the Bitstream procedure as the only viable candidate for HDTV transmission. This conclusion coincides with that of the CCSDS Red Book *Audio, Video, and Still-Image Communication Services*

The Bitstream service fills the data field of the B-PDU (Bitstream- Protocol Data Unit) with the Bitstream data supplied at user's request. Each B-PDU contains data for only one VC, identified by the VCDU-ID parameter. Each bit is placed sequentially, and unchanged, into the B-PDU data field. When the Bitstream data have filled one particular B- PDU, the continuation of the Bitstream data is placed in the next B-PDU on the same VC. Due to the delay constraints of the PDU release algorithm, if a B-PDU is not completely filled by Bitstream data at release time, some fill pattern has to be filled into the remainder of the B-PDU.

As far as the grade of service is concerned, one could use the error protection service provided by the ADTV algorithm with grade 3 service, or discard any error protection from the ADTV signal and use grade-2 service. Given the sketchy amount of information available abouthforward error correction in the ADTV algorithm we would suggest the use of the Grade-2 service in the CCSDS recommendations. Some kind of forward error correction is imperative because of the need for data sequencing along with the general need for video integrity. Therefore, Grade-2 service which adopts Reed-Soloman encoding is a logical choice. According to the minimum predicted performance of Grade-2 service, the probability that a Coded Virtual Channel Data Unit (CVCDU) will be missing is 10^{-7}. If we assume a CVCDU contains 8800 bits of data, from our simulation, about 95 macroblocks of video data (for ADTV format, a frame is formed by 90Hx60V macroblocks) will get lost in a duration slightly over one and half hour. This shouldn't hurt the quality too much in motion compensation scheme. The probability that a CVCDU contains an undetected bit error is 10^{-12}, only one bit error will occur in a transmission period over 11 hours. If this error bit occurs in video data, it won't be easy to notice the degradation. But if the error bit occurs in control data, some degree of damage is inevitable. It may therefore be desirable to provide some more protection to the control data before it enters the network. We are still looking at this particular issue.

Table 1. Performance of coding rate and PSNR using ADTV and H.261 technique
(Susie Sequence 150 frames)

	Coding Rate (Mbits/s)	STDR* (Mbits/s)	PAR**	Average PSNR	STDP*
Sequence 1	9.27	2.12	1.68	35.14	4.24
Sequence 2	15.34	2.26	1.33	37.64	4.52
Sequence 3	15.33	2.31	1.33	37.67	4.52
Sequence 4	9.39	0.61	1.65	36.83	0.81
Sequence 5	15.54	0.48	1.31	39.05	0.79
Sequence 6	15.54	0.49	1.31	38.42	0.49
Sequence 7	31.23	8.17	1.86	41.31	1.43
Sequence 8	17.27	10.72	3.37	38.77	1.68
Sequence 9	17.24	10.86	3.37	38.85	1.61

* STDR : Standard deviation of coding rate

STDP : Standard deviation of PSNR

** PAR : Peak to average ratio

frames/second : 29.97

Act. video pixels : (Luma) 1440Hx960V, (Chroma) 720Hx480V

Sequence 1 : ADTV, p=3, t=1

Sequence 2 : ADTV, p=5, t=1

Sequence 3 : ADTV, p=5, t=3

Sequence 4 : H.261, p=3, t=1

Sequence 5 : H.261, p=5, t=1

Sequence 6 : H.261, p=5, t=3

Sequence 7 : ADTV, p=10, t=1, q=4 for intra-mode frame

Sequence 8 : ADTV, p=5, t=1, q=4 for intra-mode frame

Sequence 9 : ADTV, p=5, t=3, q=4 for intra-mode frame

Fig. 1 Coding Rate using ADTV Technique

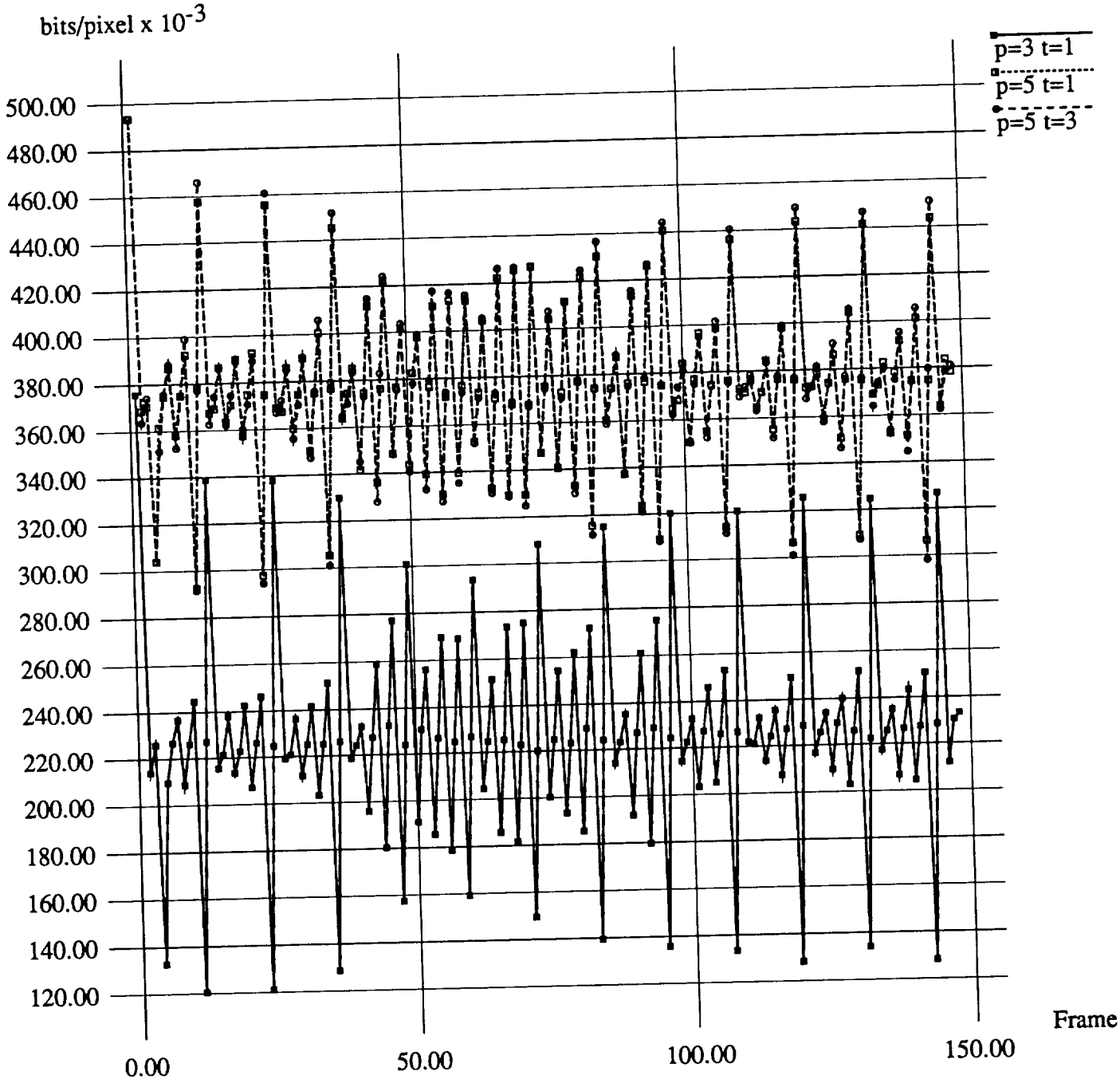


Fig. 2 PSNR using ADTV Technique

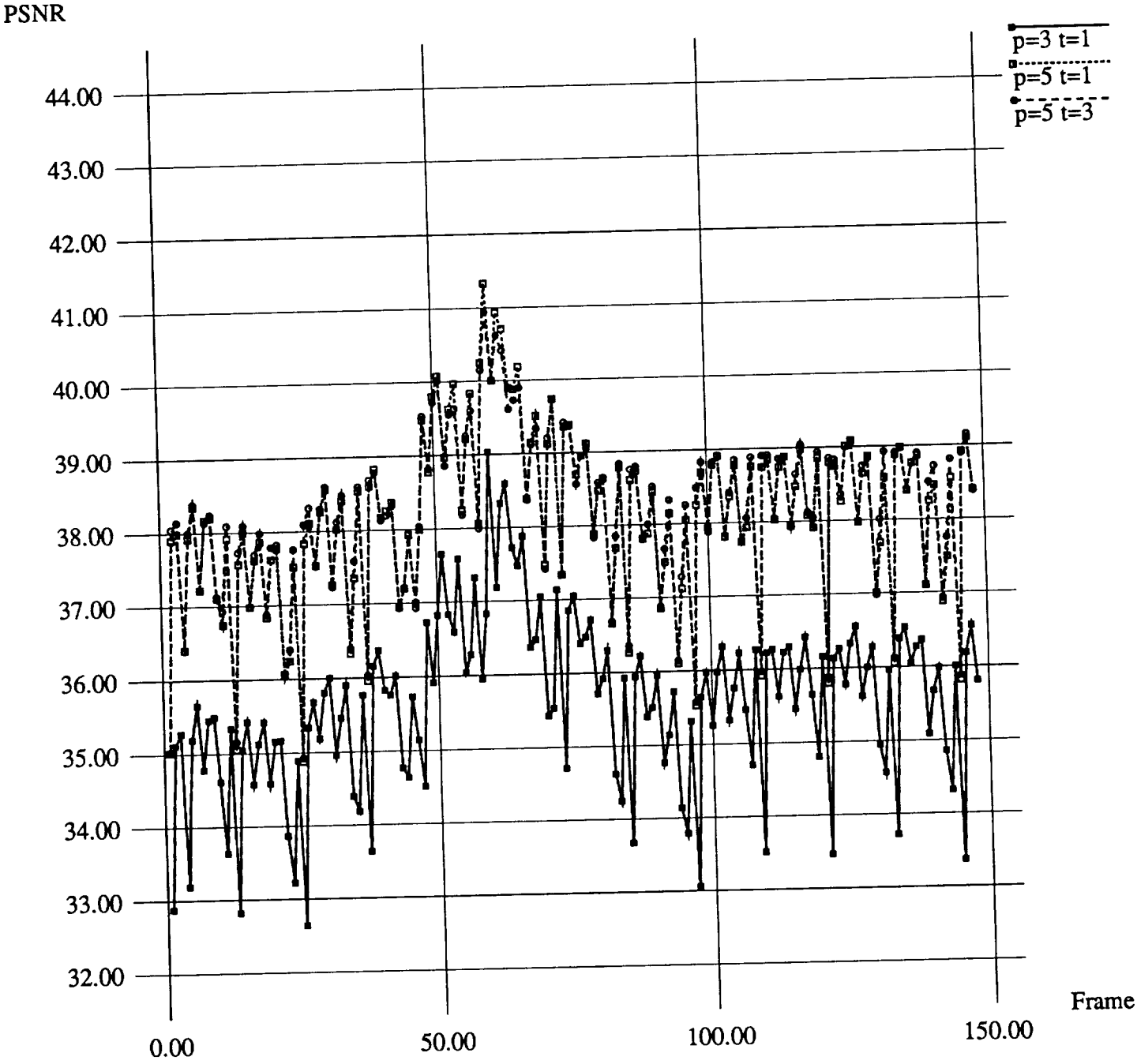


Fig. 3 Coding Rate using H.261

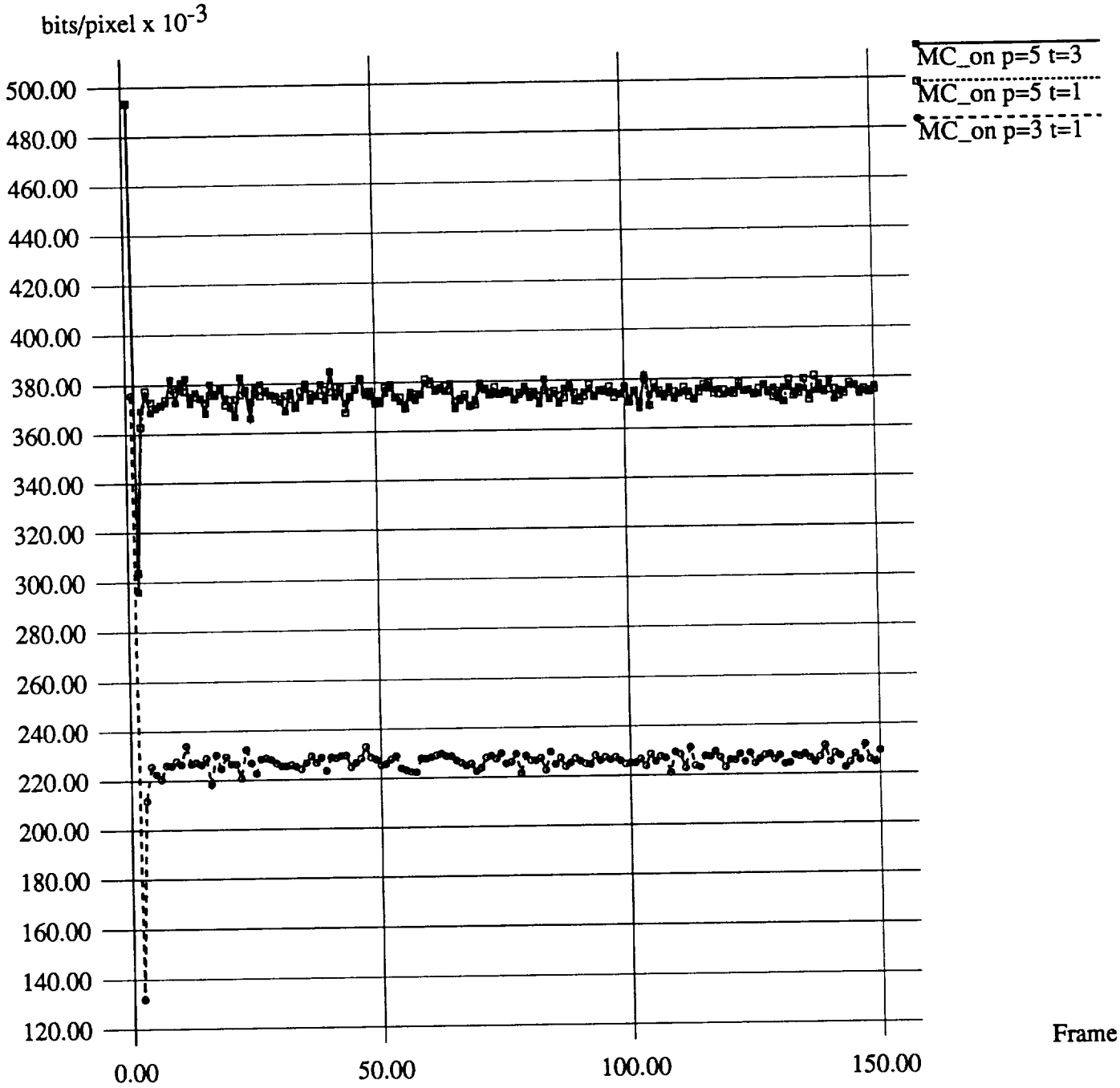


Fig. 4 PSNR using H.261

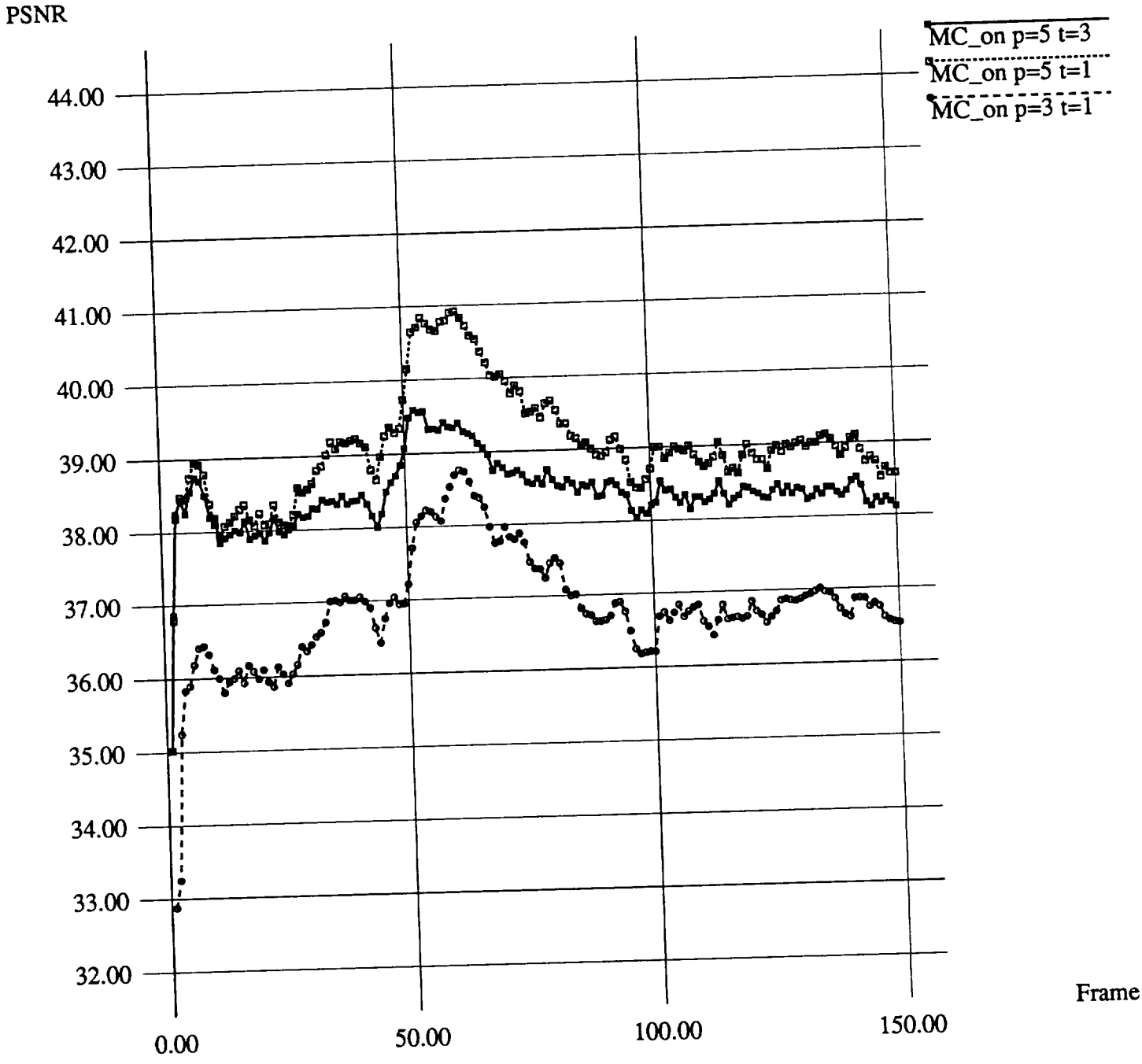


Fig. 5 Coding Rate using Modified ADTV Technique

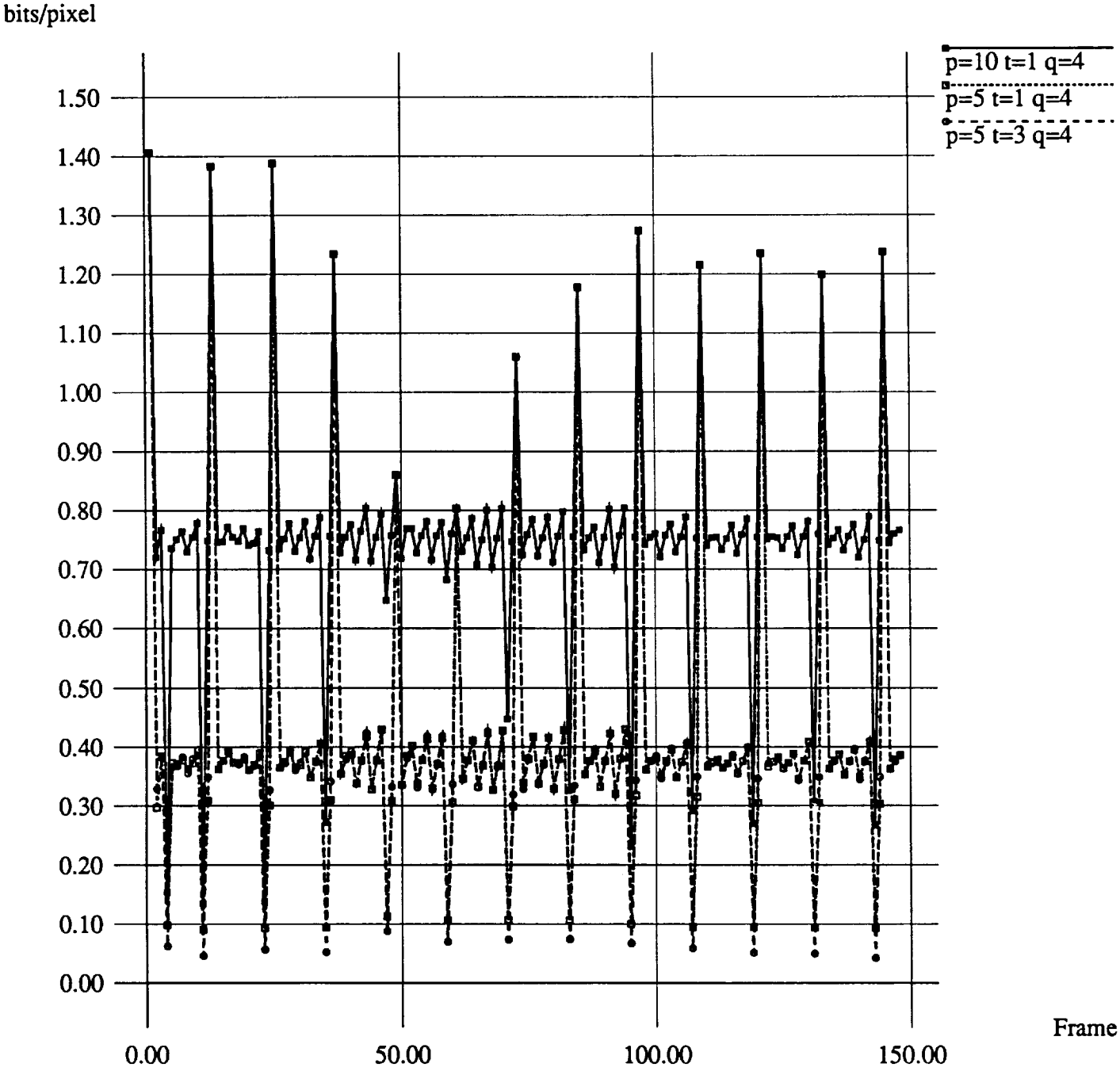


Fig. 6 PSNR using Modified ADTV Technique

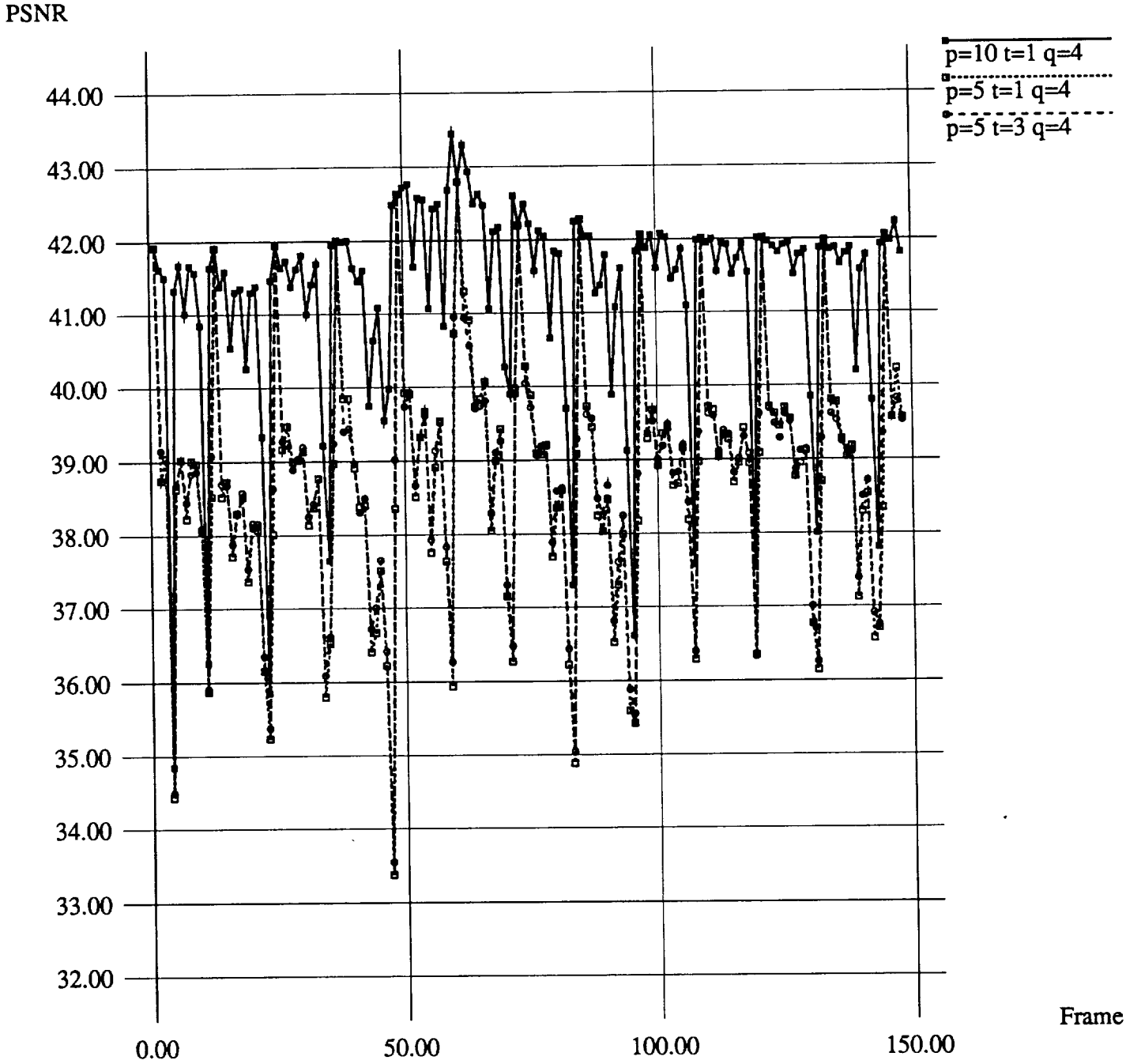


Fig. 7 Comparison of Coding Rate (p=5 t=1)

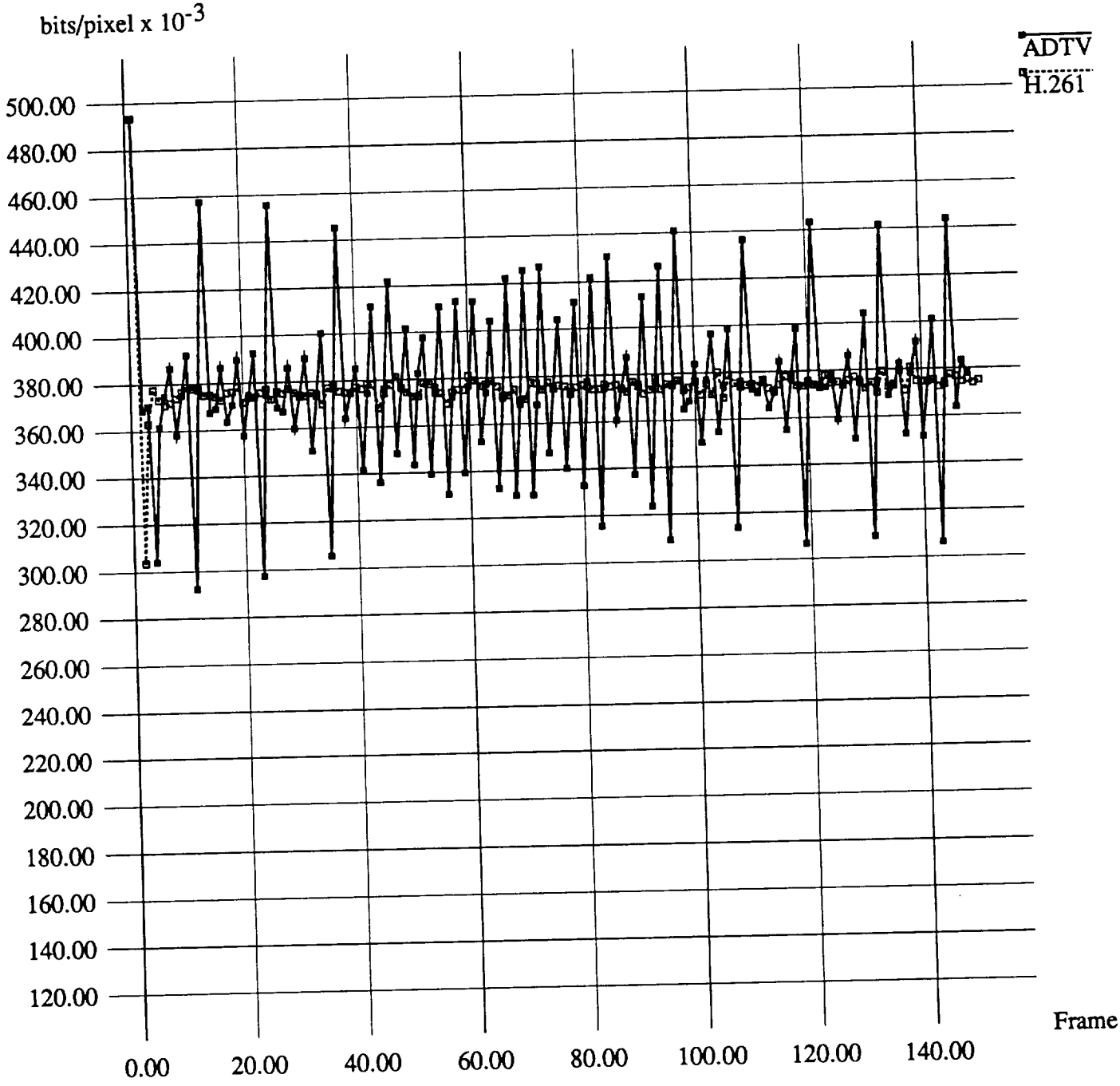


Fig. 8 Comparison of PSNR (p=5 t=1)

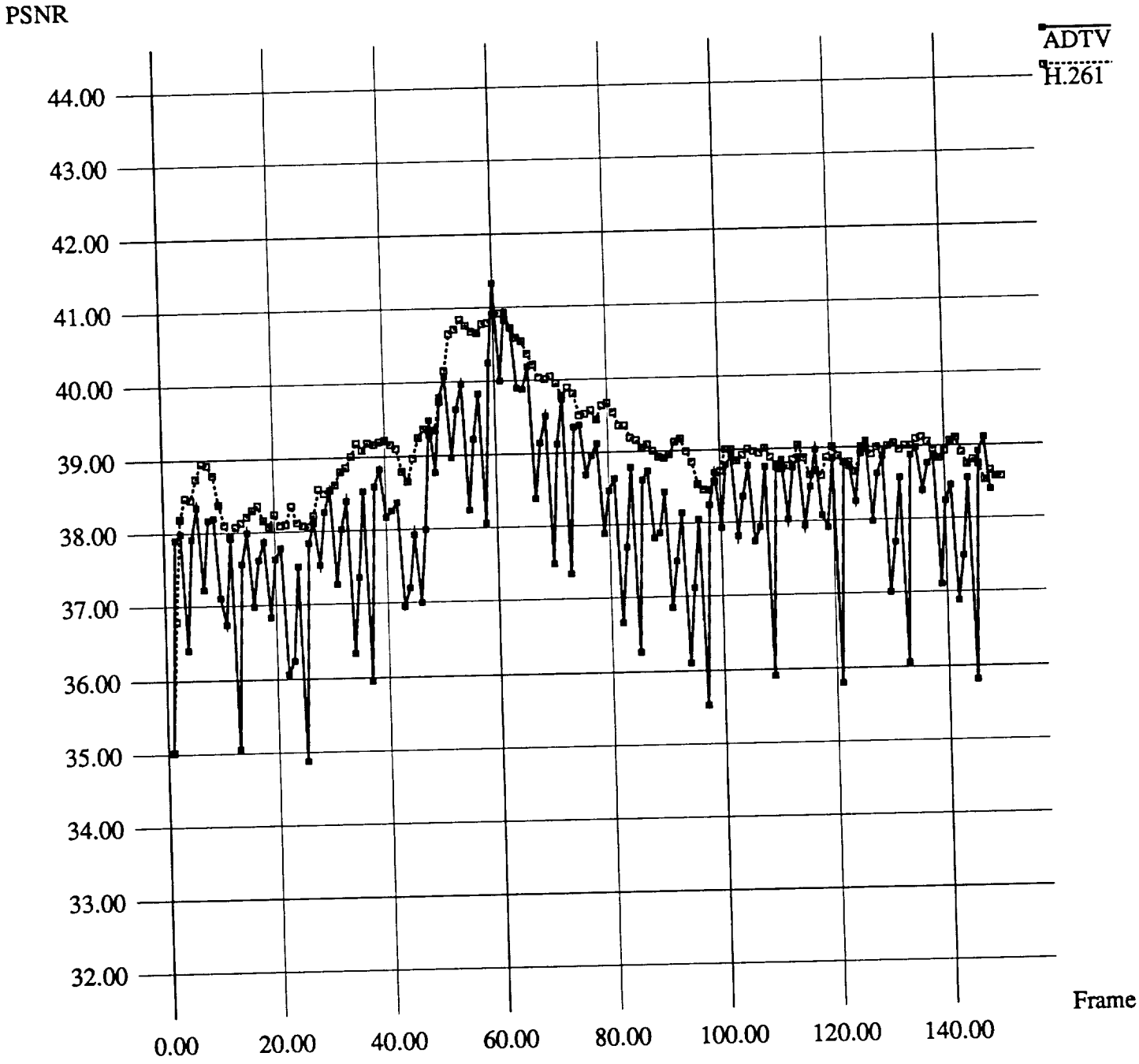


Fig. 9 Comparison of Coding Rate (p=5 t=3)

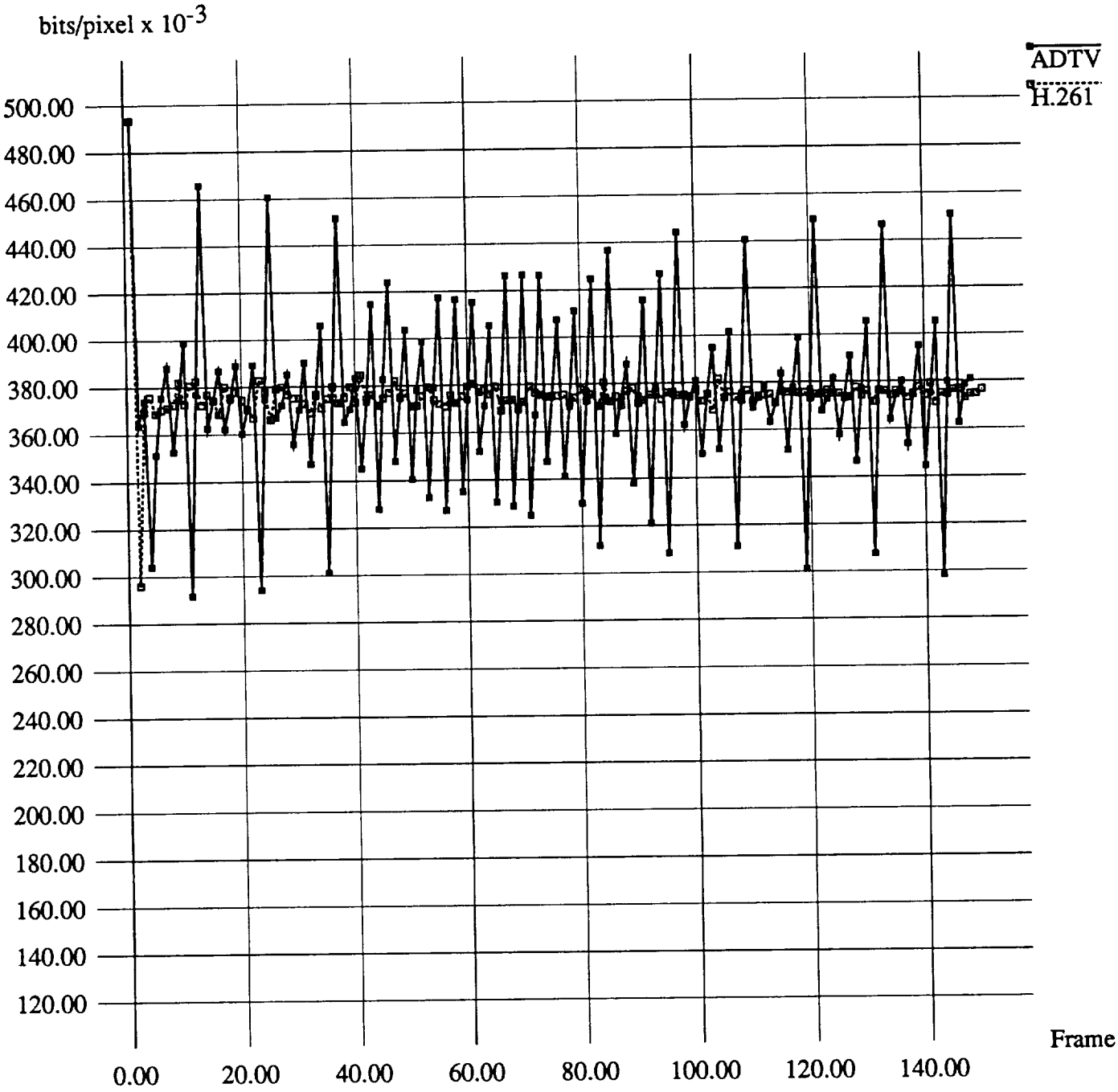


Fig. 10 Comparison of PSNR (p=5 t=3)

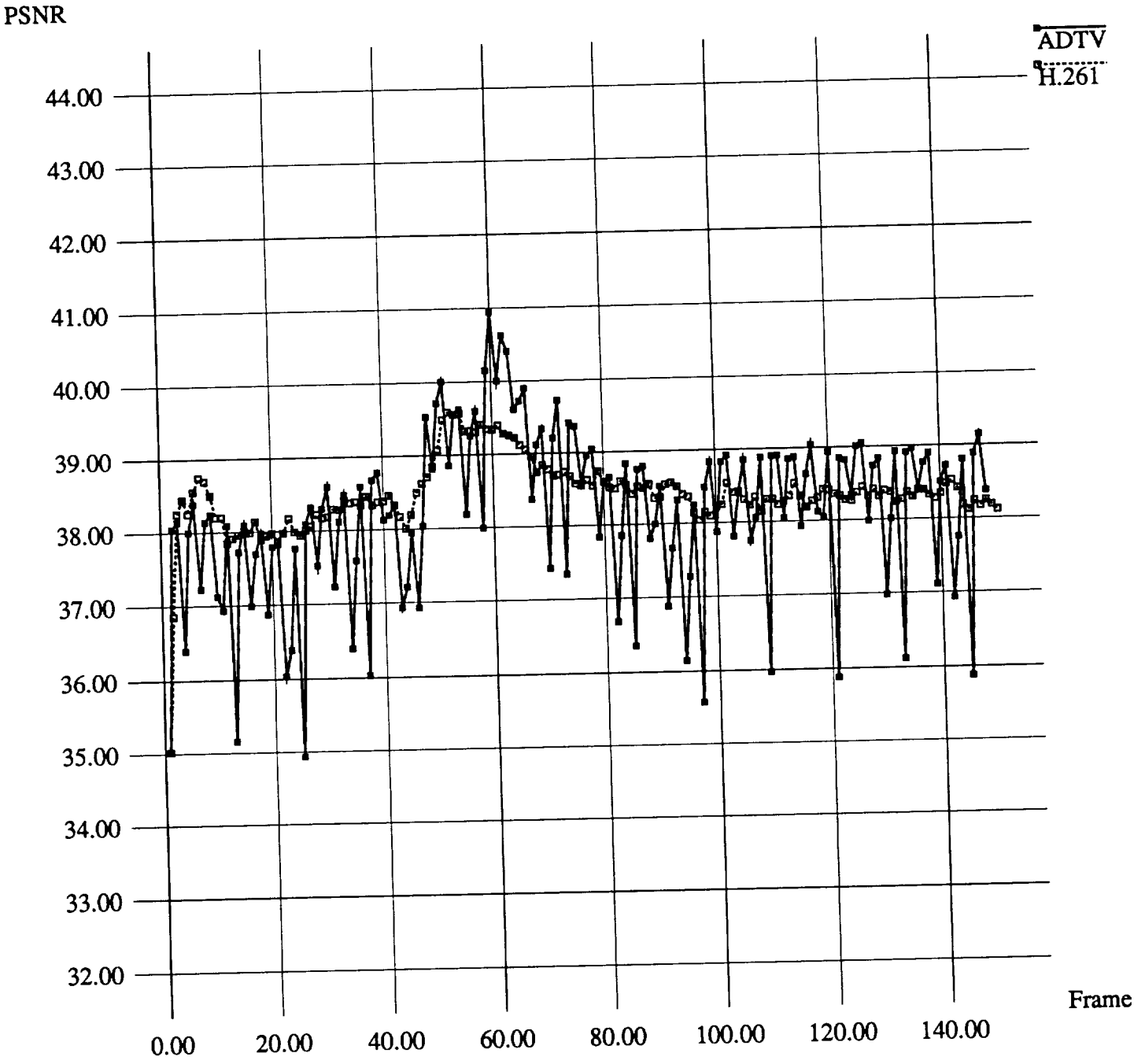


Fig. 11 Comparison of Coding Rate (p=5 t=1)

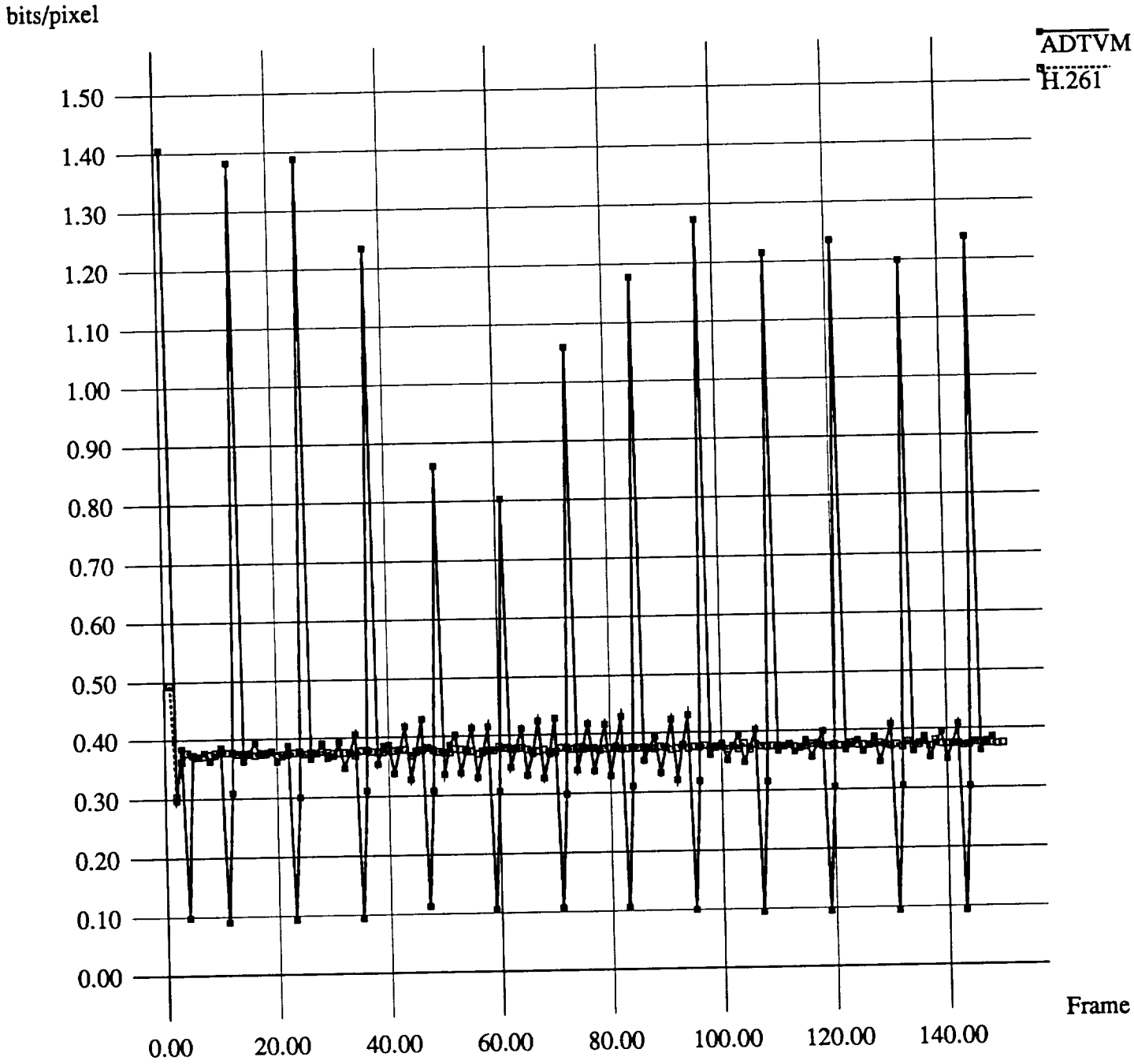


Fig. 12 Comparison of PSNR (p=5 t=1)

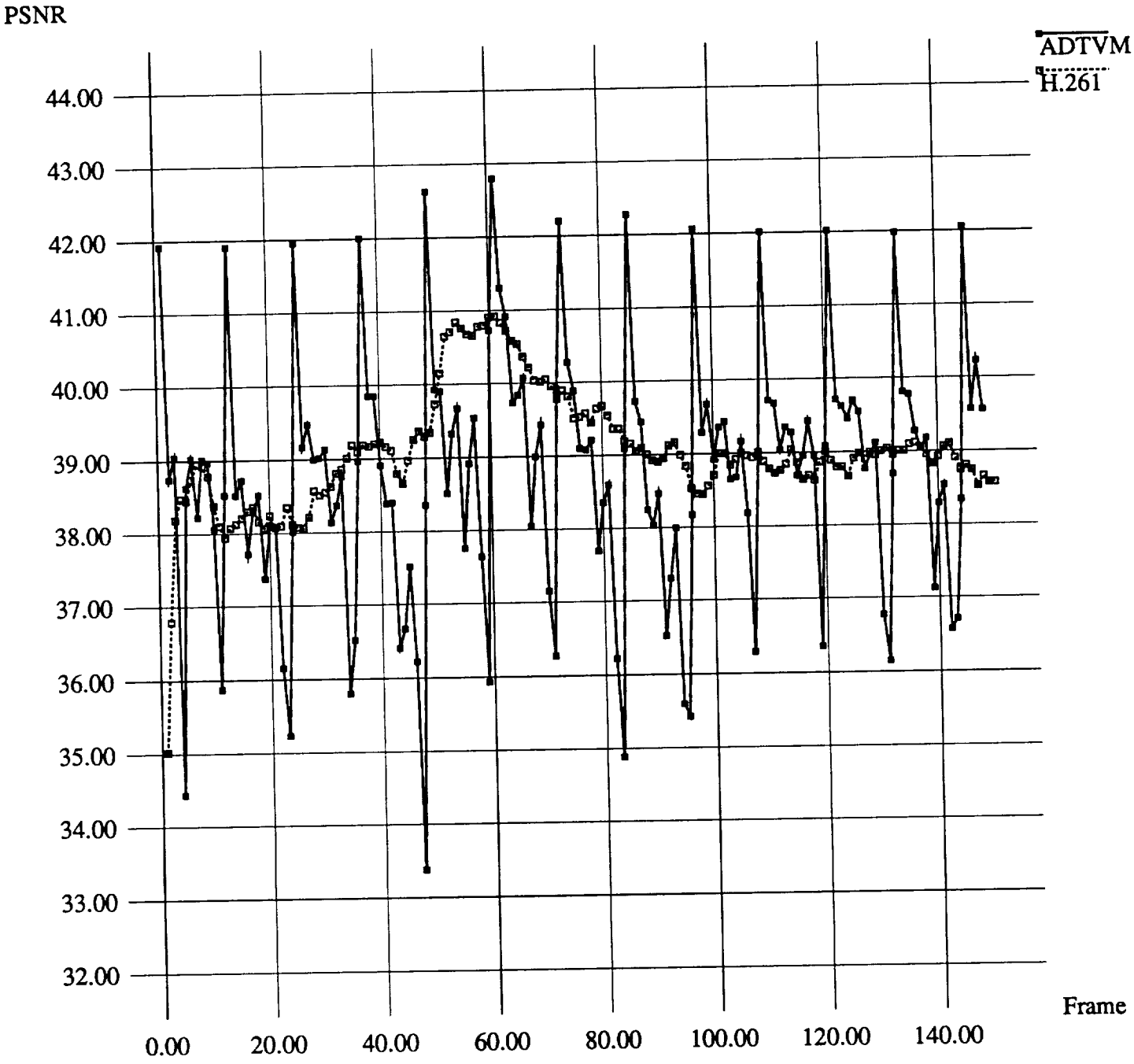


Fig. 13 Comparison of Coding Rate (p=5 t=3)

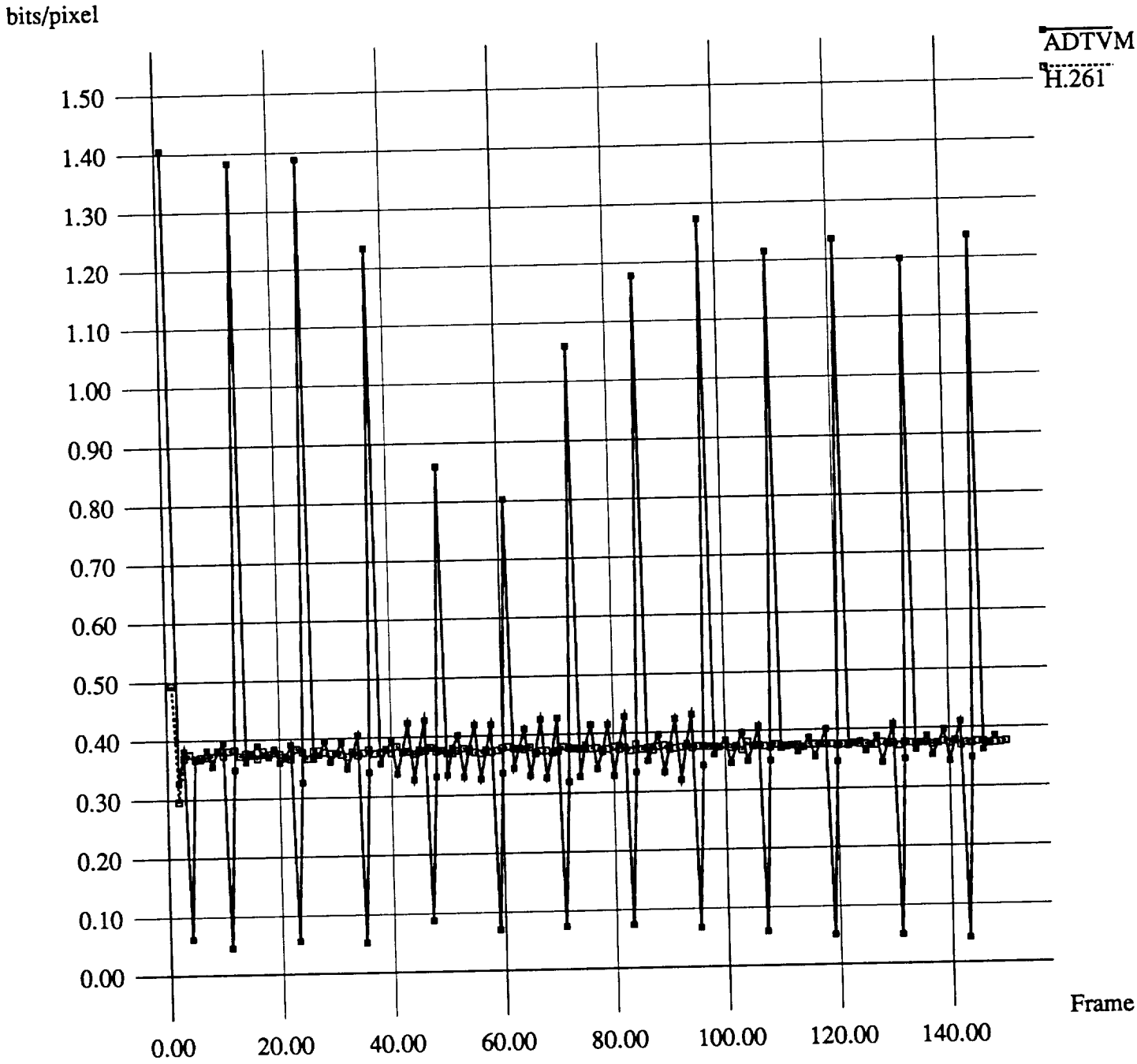
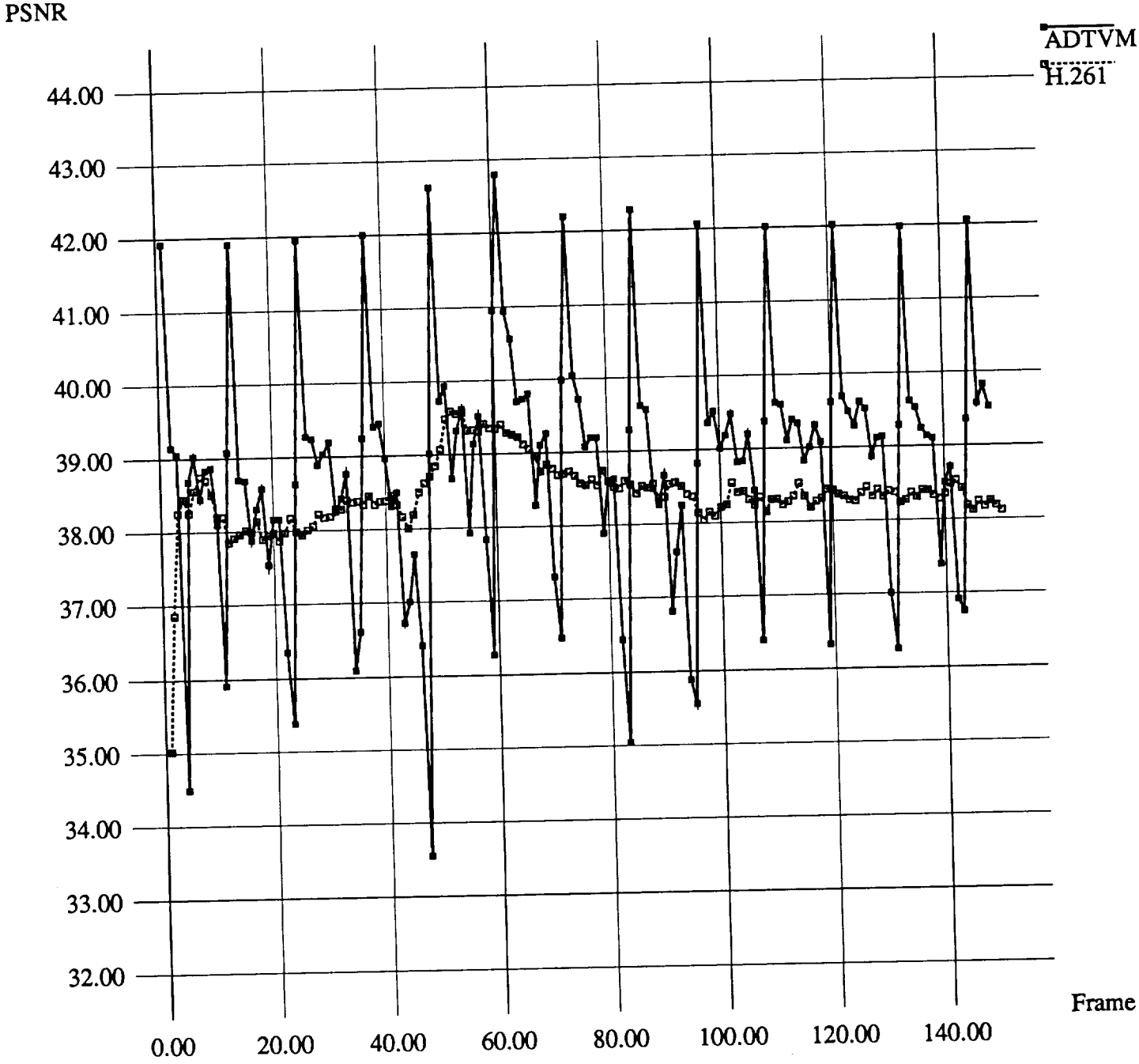


Fig. 14 Comparison of PSNR (p=5 t=3)



APPENDIX

Data Compression in Remote Sensing Applications

Khalid Sayood
Department of Electrical Engineering and
Center for Communication and Information Science
University of Nebraska - Lincoln
Lincoln Nebraska 68588-0511

1 Introduction

With the current and future availability of an increasing number of remote sensing instruments, the problem of storage and transmission of large volumes of data has become a significant and pressing concern. For example, the High-Resolution Imaging Spectrometer will acquire data at 30 meter resolution in 192 spectral bands. This translates to a data rate of 280 Mbps! The Spaceborne Imaging Radar - C (SIR-C) will generate data at the rate of 45Mbps per channel with four high data rate channels [1]. To accomodate this explosion of data there is a critical need for data compression. One can view the utility of data compression in two different ways. If the rate at which data is being generated exceeds the transmission resources, one can use data compression to reduce the amount of data to fit available capacity. Or given some fixed capacity, data compression permits the gathering of more information than could otherwise be accomodated.

In this paper, we provide a survey of current data compression techniques which are being used to reduce the amount of data in remote sensing applications. The survey aspect of this paper is far from complete, reflecting the substantial activity in this area. The purpose of the survey is more to exemplify the different approaches being taken rather than to provide an exhaustive list of the various proposed approaches. For more information on compression techniques the reader is referred to [2, 3, 4].

Compression techniques in remote sensing applications can be broadly classified into three (non distinct) categories. These are

1. Classification/Clustering
2. Lossless Compression
3. Lossy Compression

The rationale behind the classification approaches is that in a given dataset, the end user is generally interested in particular features in the data. The 'dimensionality' of these features is generally substantially less than the dimensionality of the data itself. Thus, rather than transmitting the data in its entirety, if the features are extracted on-board and transmitted this can result in a significant amount of compression. Lossless compression techniques provide compression without any loss of information. That is, the raw data can be exactly reconstructed from the compressed data. This is used when the data, or some subset of the data, is needed in exact form. In many cases, the data such as remotely sensed images, will be viewed by a human (as opposed to a machine). In these cases, distortions which are not perceptually significant can be tolerated, and lossy compression, which entails the discarding of some of the information, can be used. The utility of this approach is closely related to the amount of distortion incurred and the importance of fidelity in the particular application. The classification approaches can be viewed as a form of lossy compression. The three approaches are not mutually exclusive. For example, one may use classification as the first step with the feature vectors being losslessly encoded.

Supported by the NASA Lewis Research Center under grant NAG 3-806 and The Goddard Space Flight Center under grant NAG 5-1612.

2 Classification

If we assume an image to be composed of a small number of objects, then the most efficient form of data compression is to assign each pixel in the image to one of the objects, and then simply transmit the object labels to the ground. This idea is behind several high compression schemes which attempt to classify the pixels based on different features, and then transmit the classification map.

A technique called BLOB was introduced by Kauth et. al. [5] which uses proximity information along with spectral information for unsupervised clustering. The use of proximity information allows for greater ease in the classification of boundary pixel values, which otherwise could be classified to a set different from the adjacent regions. BLOB would be most useful in situations where objects have relatively well defined boundaries.

Another object oriented unsupervised classification scheme is described in [6]. They use what they call the *path hypothesis* for object classification. The path hypothesis assumes spatial contiguity, and spectral nearness for different pixels belonging to the same object. The spectral features of the different objects are then extracted and used to classify the object. They report an increase in classification accuracy along with a decrease in the amount of data required.

Hilbert [7] proposed a more general clustering algorithm. He proposed dividing the data into blocks, and then clustering them using an unsupervised procedure. The cluster centroids were then transmitted, along with a feature map describing the cluster to which each block belonged. This approach does not depend on the existence of well defined boundaries. Hilberts technique is a precursor to current day Vector Quantization algorithms which are discussed later.

A common precursor to classification is the transformation of the data using the Karhunen-Loeve Transform. The Karhunen-Loeve transform is used to linearly transform data

into uncorrelated coordinates. This then makes the classification task easier, as the coordinates can be clustered in a multi-dimensional space, and then classified based on their location in this space. The rows of the KL transform matrix are the eigenvectors of the correlation matrix of the data. These vectors will often be related to physical parameters. For example in [8] the first and second eigenvectors correspond to the response of the dominant surface covers. Chen and Landgrebe [8] also show that it is sufficient to send only clipped (hard limited to + – 1) eigen functions along with only a fraction of the coefficients to obtain significant classification accuracy. They therefore propose the use of this scheme aboard the HIRIS instrument.

3 Lossless Compression

Lossless compression, as the name implies, consists of reduction in the amount of data without sacrificing the fidelity of the data. The earliest known lossless compression technique of the technological age is probably the Morse code. In the Morse code, letters that occur often such as E are coded using short symbols, while letters that occur relatively infrequently such as Z, are represented by long symbols (a single dot for E and dash dash dot dot for Z). This idea (albeit in more sophisticated form) is at the heart of most lossless compression schemes. In 1948 Claude Shannon defined the amount of information contained in the event X as $log_a \frac{1}{P(X)}$ [9], where $P(X)$ is the probability of the event X and a is the base of the logarithm. If $a = 2$ the unit of information is bits. If we define X^n to be the sequence of observations $(X_0, X_1, \ldots, X_{n-1})$, then the *entropy* of the source generating the sequence is defined as

$$H(S) = \lim_{n \to \infty} G_n$$

where

$$G_n = \sum P(X^n) log_2 \frac{1}{P(X^n)}$$

Shannon [9] showed that the minimum average rate at which the output of the source S can be encoded is H(S) bits/symbol. If the source outputs $\{X_i\}$ are independent then the expression for entropy reduces to

$$H(S) = G_1 = \sum P(X_i) \, log_2 \frac{1}{P(X_i)}$$

Given a sequence of independent observations, Huffman [10] developed an algorithm which provides a variable length code which gives an average coding rate R, where $H(S) \le R \le H(S) + 1$. The algorithm assigns shorter codewords to more probable symbols and longer codewords to less probable symbols *a la* Morse. Another technique which operates on sequences rather than individual letters is *Arithmetic Coding*. The Arithmetic coding algorithm guarantees an average coding rate R where $H(S) \le R \le H(S) + \frac{2}{n}$, n being the length of the sequence. If the statistics of the sequence change with time, these techniques will suffer some degradation. To combat this several adaptive coding techniques have been proposed including dynamic Huffman coding [11], adaptive arithmetic coding [12] and the Rice algorithm [13]. The Rice algorithm has been shown to be optimal under some widely available conditions [14], and has been implemented in a VLSI chip which can process 20 M-Bytes per second [15].

If the observations are not independent then the code designed using the first order probabilities $P(X_i)$ is only guaranteed to be within one bit of G_1 which may be substantially greater than $H(S)$. Because of this fact lossless compression consists of two steps; decorrelation, and coding. The first step can be seen as an 'entropy reduction' step in which the redundancy or correlation of the data is removed (reduced). This results in another sequence which has a first order entropy G_1 which can be significantly lower than the first order entropy of the original sequence. Now if a variable length code is designed using the first order probabilities of the decorrelated data, this will result in a lower rate/higher compression. Consider for example the following sequence

12345432123212345432345

estimating the first order probabilities from the sequence we obtain

$$P[1] = P[5] = \frac{3}{23} \; ; \; P[2] = P[3] = \frac{6}{23} \; ; \; P[4] = \frac{5}{23}$$

which gives a value for G_1 of 2.25 bits/sample. It is obvious from looking at the data that it possesses some definite structure. Some of this structure can be removed by storing consecutive differences. The original data can be reconstructed (without loss) by simple addition. The difference data is

1 1 1 1 – 1 – 1 – 1 – 1 1 1 – 1 – 1 1 1 1 – 1 – 1 – 1 1 1 1

The difference can be represented using a binary alphabet, so the coding rate can immediately be lowered to one bit/sample. To see what the value of G_1 is we first compute the first order probabilities as $P[1] = \frac{14}{23}$, $P[-1] = \frac{9}{23}$, which gives an entropy of .96 bits/sample. In this particular case the gain of 0.04 bits per sample may not be worth the additional complexity required for a variable length code. Notice that in this case the compression was obtained mainly

due to the decorrelation step. Because of this, research in lossless compression is focusing more and more on the development of better decorrelation algorithms. An idea of how much decorrelation gain is available can be obtained by looking at the conditional entropy.

In the example given above, the data was one dimensional so the *prediction* used to generate the difference or *residual* data was also one dimensional. In the case of remotely sensed images, the data is generally three dimensional: two spatial dimensions and a spectral dimension. In these cases, it would seem reasonable to use prediction based on all three dimensions. Chen et. al. [16] compute the theoretical advantages to be gained from using prediction based on all three dimensions. They show that while there is some advantage to be gained from using more than one dimensional prediction, the increase in compression is small. However, if the increase in complexity of going from one to two or three dimensions is acceptable (and it can be argued that the increase in complexity is minimal), it would seem reasonable to use multi-dimensional prediction to decorrelate the data.

A somewhat different approach is adopted by Memon et. al. [17, 18]. They reason that in an image the correlations may be maximum in the vertical, horizontal, or diagonal direction depending on the object being imaged. Therefore, one should use whichever pixel gives the most decorrelation for prediction. They therefore develop the concept of prediction (or scanning) trees for performing the decorrelation. The drawback with this approach when coding single images is that the cost of encoding the prediction tree may eat up any savings due to better decorrelation. In the case of multispectral images, because the same prediction tree can be used to code a large number of bands, the relative cost of encoding the prediction tree is small enough not to overwhelm the savings obtained via this approach [19].

In all that we have discussed above, we have taken a rather general view of the lossless compression problem. When faced with a specific problem, one can often come up with a simpler more efficient solution. Consider the problem of encoding the output of a spectrometer. A general algorithm such as the Rice algorithm will do a nice job of encoding the output of the spectrometer. However, given the very special structure of the data (the data looks like a noisy decaying exponential) one can come up with simpler techniques as in [20] which are simpler and give better performance. Similarly Stearns et. al. [21] develop a lossless compression scheme tuned to the peculiarities of seismic data. When using application specific algorithms, the user should be aware of the fact that if the data sequence deviates from the assumed structure, this may result in performance loss.

Finally, lossless coding can be used in conjunction with other techniques. Several schemes in the literature use lossless compression as the second stage, where the first stage is feature extraction or lossy compression [23, 23, 24].

4 Lossy Compression

In many applications, loss of information which is not perceptually significant can be easily tolerated. In fact in certain cases, such as processing of SAR data [25], the 'information' lost may actually be the noise. In these cases, it makes sense to use lossy compression techniques which provide much higher compression than the lossless techniques. However, before we extoll the virtues of various lossy compression techniques, one should keep in mind the importance of carefully picking the distortion measure. Most of the compression schemes described here use the mean squared error (or some variant) as the distortion measure. The mean squared error is defined as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

where x_i is the original data value and $\hat{x}_i$ is the *reconstructed* (compressed and then decompressed) value. Note that this is an *average* measure therefore it will spread out the error effects at any one location. Under this measure, a large error in one sample value with no or little error in the other N-1 sample values may be equivalent to small errors in all N sample values. If the application requires that each sample value be represented within some tolerance, then the MSE is probably not the distortion measure that should be used.

4.1 Quantization

The heart (and sometimes the totality) of most lossy compression schemes is the quantization process. Quantization is a many to one mapping from a possibly infinite set to a finite set. The input to the quantizer can be a scalar, in which case the quantizer is called a scalar quantizer, or a vector in which case the quantizer is called a vector quantizer (VQ). The scalar quantizer is simply a concatenation of an A/D and a D/A. A simple A/D is shown in Figure 1. Assuming $\Delta = 1$, in this A/D if the input falls in the range (0,1], the output is the codeword 10, if the input falls in the range $(-\infty,-1]$ the output is the codeword 00, and so on. The D/A takes the codeword produced by the A/D and generates a real value corresponding to the interval represented by the codeword. In our simple example if the codeword 00 is received the D/A will put out a value of -1.5. The input/output map for this quantizer (A/D-D/A combination) is shown in Figure 2. Figures 1 and 2 describe a two bit *uniform* quantizer. If the stepsize Δ is not constant for the different intervals, the quantizer is called a *non-uniform* quantizer. Given information about the statistics of the input signal, Max [26] and Lloyd [27] have developed algorithms for the design of optimum uniform and non-uniform quantizers for memoryless sources. Kwok and Johnson [28] use a two bit quantizer designed for Gaussian data to code SAR data from the Magellan mission. The SAR data is originally at 8-bit resolution, so the

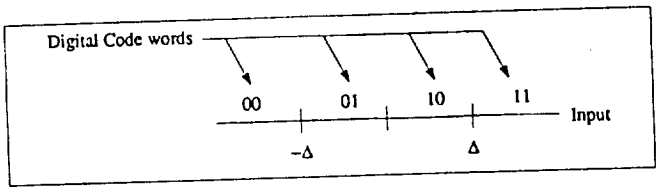


Figure 1. A two bit A/D

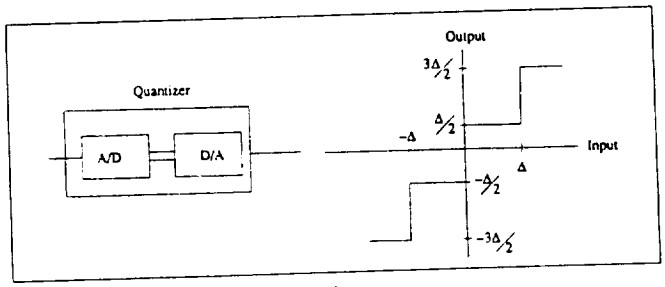


Figure 2. A two bit uniform quantizer

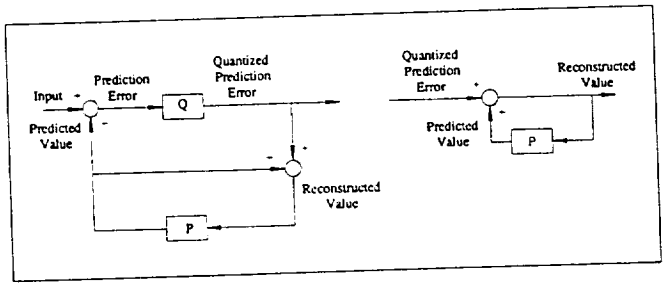


Figure 3. DPCM structure

compression ratio is 4:1. To accomodate the rather large dynamic range of the SAR data, the quantizer is adapted on a block by block basis, using the average signal magnitude. The signal magnitudes in a block are used to compute a threshold value which is used in place of Δ in Figure 1. The output of the D/A are the optimum values for a Gaussian input with variance of one multiplied by the computed threshold.

The SIR-C [1] uses 8 bit uniform quantization followed by a feature which allows it to reduce the number of bits per sample to facilitate the acquisition of more samples. Data compression thus allows the acquisition of more data at the cost of reduced resolution.

In some cases it might be more efficient to quantize some function of the data rather than the data itself. Dubois et. al. [29] compress the output of an imaging radar polarimeter by first obtaining the Stokes matrix from the scattering matrices. Four Stokes matrices from contiguous pixels are added to form one four-look Stokes matrix. The elements of the four-look Stokes matrix are then quantized. The advantage to this approach is that the elements of the Stokes matrix have certain well defined properties which can be used in the quantization process of the Stokes matrix.

4.2 DPCM

The relationship between the variance of the input to the quantizer and the *MSE* can be given by the following relationship,

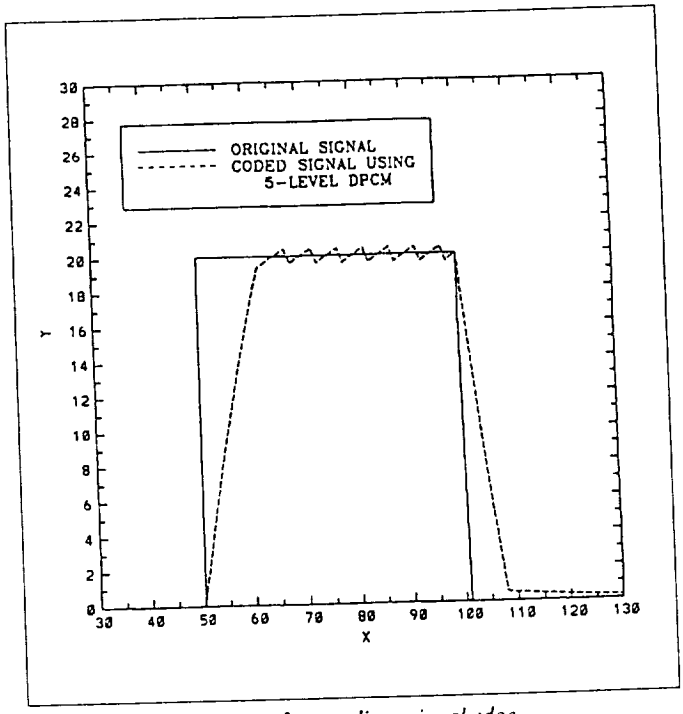


Figure 4. DPCM coding of a one dimensional edge

$$MSE = \varepsilon_*^2 \, 2^{-2R} \, \sigma_x^2$$

where ε_*^2 depends on the input probability density function, and R is the number of bits/sample. As can be seen from this expression, the *MSE* is proportional to the input signal variance. Therefore, if we could reduce the input signal variance this would lead to a reduction in the *MSE*. (It should be noted that the operations to remove the redundancy could also change the input pdf which may diminish the benefits of a reduced variance.) This is the motivation for a class of lossy compression schemes known as Differential Pulse Code Modulation (DPCM) schemes. DPCM schemes remove redundancy in the source sequence by using the correlation in the source sequence to predict ahead. The predicted value is removed from the signal at the transmitter and reintroduced at the receiver. The prediction error, which has a smaller variance than the input signal is then quantized and transmitted to the the receiver. A block diagram of a DPCM system is shown in Figure 3. This technique is used in the coding of the SPOT satellite's panchromatic band.

While DPCM coding performs well in quasi-stationary regions of an image, it does a poor job in edge regions. The reason for this is that the prediction in DPCM uses the previous *reconstructed* pixels. In an edge region, the prediction error is quite large. Therefore, the input to the quantizer lands in one of the outer regions ((-∞,-1],[1,∞) in our example). The quantization error can therefore be quite large. This is fed back via the prediction process into the coding of the next pixel, and so on causing a smearing of the edges. This process is demonstrated on a one-dimensional 'edge' in Figure 4. This problem can be overcome by using recursive-

Figure 5. Original aerial view of Omaha

Figure 6. DPCM coded Omaha image at 1.4 bpp

ly indexed quantization [30, 31] which avoids the large quantization error problem by operating the quantizer in two different modes. Whenever the input to the quantizer falls in the external regions, the quantizer switches into a recursive mode, and the quantization error is requantized until the error falls within some predetermined tolerance. This approach not only prevents large quantization errors from propagating through the coded sequence, it also guarantees that the error *per pixel* will be less than a pre-determined value. To show how well this scheme works, we code the aerial view of Omaha shown in Figure 5. The compressed (and decompressed) image coded using the DPCM scheme described above at a rate of 1.4 bits per pixel is shown in Figure 6. Note that while there is an overall increase in 'blurriness' the distortion introduced does not blur the edges.

While the DPCM structure removes substantial amounts of the redundancy from the data stream, it should be remembered that the prediction process in the DPCM structure is linear, and can therefore remove only those redundancies which are expressed as linear processes. For example, a slowly varying sequence 1 2 3 4 5 4 3 3 4 5 6 7 7 7 6 has redundancies that can be modeled by a linear process. However, we can easily come up with sequences that have redundancies that can not be characterized by a linear process such as 4 24 15 19 4 24 15 19 .... This fact has been used by some to improve the data compression by making use of this redundancy for code selection [32], and by others for providing error protection [33].

4.3 Vector Quantization

Until now we have been talking about quantization as a scalar process, however, the basic idea of quantization can easily be extended to the vector case. Scalar quantization can be viewed as a partition of the real number line, with the A/D doing the partitioning, and the D/A providing a representative value for each partition. Similarly, vector quantization can be seen as a partitioning of multidimensional space. While conceptually the problems of scalar and vector quantization approaches are very similar, the practical problem of designing vector quantizers is significantly more difficult. Two somewhat different approaches have been taken towards the design of vector quantizers. The first is a clustering approach similar to the Hilbert technique [7]. In this approach [34], a training sequence is used to identify the regions in multi-dimensional space where the data seems to cluster. The quantizer outputs are the centroids of these clusters, and the partitions are the nearest neighbor partitions of these centroids. An example of a two dimensional vector quantizer is shown in Figure 7. The VQ in Figure 7 contains 4 output levels, or codewords. Thus the size of each codeword is two bits. But each output level corresponds to the coding of two input samples, therefore, the number of bits per sample is one. In general, given the dimension of the vector d and the number of bits per sample R, the size of the vector codebook is 2^{Rd}. Notice, that this means an exponential increase in the size of the codebook with dimensionality and rate. For example, given $d = 12$ and $R = 2$, the size of the codebook would be $2^{24} = 16777216$! This represents an enormous expense in storage and computing resources. Thus the rate-dimension product provides a limitation on the clustered VQ designs. Fortunately, a lot can be done at low rate-dimension products. For more moderate rate-dimension products a number of somewhat more structured VQ algorithms have been developed [35]. Chang et. al. [25] report the use of a tree-structured VQ on Seasat SAR imagery with favorable results. As the codebook of the VQ is obtained by training, it is important that the data in the training set be representative of the data in the test set. If this is not the case, there can be

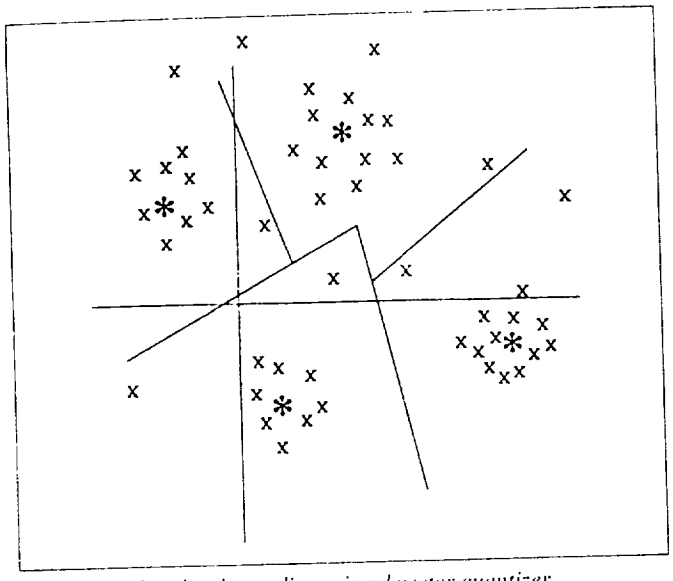


Figure 7. A four level, two dimensional vector quantizer

significant degradation in the data that is not typical of the training set [25]. The Omaha image of Figure 5 is coded using a clustered VQ at 0.5 bits per pixel. The result is shown in Figure 8. The VQ dimension was 16 (4X4 blocks) and this is evident from the coded image in Figure 8, where there is a noticeable amount of blockiness. The blocks that lie on the edges of objects in the image clearly distort the edges. The VQ codebook was obtained using another aerial image. We can improve the performance of this algorithm by increasing the rate and/or by generating the codebook from an image which more closely resembles the image being coded. In Figure 9 we have the Omaha image coded at 1 bit per pixel using a codebook generated using the Omaha image itself. There is substantial improvement in the quality, though there is still some distortion in the lower quarter of the picture. It should be noted that the use of the image to generate the codebook is generally not realistic.

Vector Quantization is also used by Gupta and Gersho for the coding of Landsat TM images [36]. They use a vector DPCM system with vector quantization in the spatial domain, and predictive encoding in the spectral domain. A variation of predictive VQ is also used by Giusto [37] for the compression of multispectral images.

The rate-dimension product constraint on vector quantizers can be lifted by making the vector quantizer more and more structured. Of course, as the VQ acquires more and more structure of its own, it is less and less responsive to structure in the data. The most structured vector quantizers are those based on a multi-dimensional lattice [38]. While these quantizers do an excellent job of quantization, they cannot at the same time perform the redundancy removal operation performed by the clustered VQs. They therefore have to be used in conjunction with other techniques to provide compression [39, 40].

4.4 Transform Coding

Most of the techniques we have talked about operate in the data domain, i.e. without any transformation. There is a large class of compression techniques that operate on a transformed version of the data. They are called transform coding techniques. The idea behind transform coding is to transform the data in such a way as to compact most of the energy (and information) into a few coefficients. These coefficients can then be coded, while other coefficients can be discarded thereby achieving data compression. The most efficient transform from the compaction point of view is the Karhunen-Loeve [2] transform. However, the Karhunen-Loeve transform is data dependent which makes it impractical for most compression applications. The best alternative to the

Figure 8. VQ coded Omaha image at 0.5 bpp

Figure 9. VQ coded Omaha image at 1.0 bpp

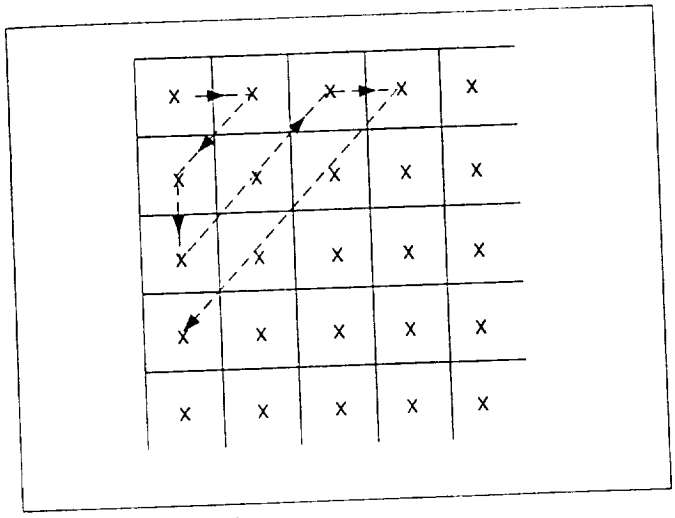


Figure 10. Zig zag ordering

Karhunen Loeve transform is the Discrete Cosine Transform (DCT). This is a real, separable, unitary transform that is the basis for an image compression standard [41]. Because of its popularity in image compression various fast algorithms have been proposed for its implementation [42, 43].

The ESA Huygens Titan Probe to be launched by the Cassini Orbiter will use the DCT for compressing the image data acquired during its descent through Titan's atmosphere. The images of size 256X256 will be divided into 8X8 blocks. These blocks will be transformed and the transform coefficients reordered using the zigzag ordering shown in Figure 10. The ordered coefficients will then be blocked into substrings of four coefficients each. Substrings with all coefficient values below a specified threshold will be deleted while the remainder will be quantized using scalar quantizers. Details can be found in [44, 45].

To see the artifacts introduced by DCT coding we have coded the Omaha image at 0.5 bits per pixel and 1 bit per pixel as shown in Figures 11 and 12. Note the substantial block artifacts in Figure 11 which have been reduced to a large extent in Figure 12. However even in Figure 12 one can see significant distortion in edge regions.

An adaptive version of DCT was also considered by Chang et. al. [25] for the compression of Seasat SAR imagery. They compare the DCT technique with a VQ technique and decide in favor of the VQ technique based on complexity issues. With the wide acceptance of the DCT as an image compression standard, the complexity issue may no longer be relevant, as more and more manufacturers are bringing hardware implementations of the DCT to the market.

5 Conclusions

As can be seen from this discussion, there is a substantial amount of on-going activity in the area of data compression for remote sensing applications. This will only increase as there is more and more need for data compression. However, ther are several areas of research which have not been addressed in any significant way.

There is a need for the development of better distortion measures which can be then used to develop more sophisticated compression algorithms. It is possible that rather than a single distortion measure, a set of distortion measures will be needed for different applications. The development of such measures, and algorithms utilizing these measures, require close cooperation between data compression specialists and the scientists and engineers who are the end-users of the data obtained through remote sensing.

The multi-dimensional (spatial and spectral) nature of the data has not really been thoroughly explored (except in the

Figure 11. DCT coded Omaha image at 0.5 bpp.

Figure 12. DCT coded Omaha image at 1.0 bpp.

classification approaches). With the development and deployment of high spectral resolution instruments, this particular aspect of remotely sensed data will become more important. Compression schemes which take advantage of this fact need to be developed. An analogy could be drawn with the development of compression algorithms for video as opposed to still images. However, the algorithms developed for video cannot be directly applied to high spectral resolution image data sets, as the differences that occur between frames of a video sequence are not the same as the differences that occur between different spectral images. It would seem that VQ approaches such as [36] would provide possible solutions. The rate-dimension constraints in clustering VQ could be avoided by the use of Lattice VQ techniques. Another approach described in [46] is to use a two step strategy, in which the first step is used to model the data in the spectral direction. The resulting models are then treated as a vector image for compression in the spatial directions. Beyond this, however, there is a need for the development of three dimensional approaches, both to model the data, and develop compression algorithms.

6 Acknowledgement

The author would like to thank the reviewers for their careful reading of the manuscript and for their suggestions. The DCT coded images were obtained using software from the Independent JPEG foundation. The VQ images were provided by Dr. Nasir Memon.

References

[1] B.L. Huneycutt. Spaceborne Imaging Radar-C Instrument. *IEEE Transactions on Geoscience and Remote Sensing*, pages 164-169. March 1989.

[2] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[3] M. Rabbani and P.W Jones. *Digital Image Compression Techniques*, volume TT7 of *Tutorial Texts Series*. SPIE Optical Engineering Press, 1991.

[4] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.

[5] R.J. Kauth, A.P. Pentland, and G.S. Thomas. BLOB: An Unsupervised Clustering Approach to Spatial Preprocessing of MSS Imagery. In *Proceedings Int Symp Remote Sensing Environ*, pages 1309-1317, 1977.

[6] H. Ghassemian and P.A. Landgrebe. Object Oriented Feature Extraction Method for Image Data Compaction. *IEEE Control Systems Magazine*, pages 42-48, June 1988.

[7] E.E. Hilbert. Cluster Compression Algorithm - a Joint Clustering Data Compression Concept. Technical Report JPL Publication 77-43, NASA, 1977.

[8] C-C.T. Chen and D.A. Landgrebe. A Spectral Design System for the HIRIS/MODIS Era. *IEEE Transactions on Geoscience and Remote Sensing*, 27:681-686, November 1989.

[9] C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:398-403, July 1948.

[10] D.A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40:1098-1101, 1951.

[11] R.G. Gallagher. Variations on a theme by Huffman. *IEEE Transactions on Information Theory*, IT-24(6):668-674, November 1978.

[12] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley and Sons Inc., 1991.

[13] R.F. Rice. Some practical universal noiseless coding techniques. Technical Report JPL Publication 79-22, NASA, 1979.

[14] P-S. Yeh, R.F. Rice, and W. Miller. On the Optimality of Code Options for a Universal Noiseless Coder. Technical Report JPL Publication 91-2, NASA, February 1991.

[15] J. Venbrux and N. Liu. A Very High, Speed Lossless Compression/Decompression Chip Set. In *Data Compression Conference*, 1991.

[16] T.M. Chen, D.H. Staelin, and R.R. Arps. Information Content Analysis of Landsat Image Data for Compression. *IEEE Transactions on Geoscience and Remote Sensing*, GE-25:499-501, July 1987.

[17] N.D. Memon, S.S. Magliveras, and K. Sayood. Prediction trees and lossless image compression - an extented abstract. In Reif J.H. Storer J.A., editor, *Proceedings of the Data Compression Conference*, pages 83-92. IEEE Computer Society Press, 1991.

[18] N.D. Memon, K. Sayood, and S.S. Magliveras. Lossless Image Compression Using Efficient Scans. In *26th Annual Conference on Information Sciences and Systems*, March 1992.

[19] N.D. Memon. *Image Compression Using Efficient Scanning Patterns*. PhD thesis, University of Nebraska - Lincoln, August 1992.

[20] K. Sayood and M.C. Rost. A Robust Compression Scheme for Low Bit Rate Telemetry - Test Results with Lunar Data. In *Scientific Data Compression Workshop*, pages 237-250, May 1988.

[21] S.D. Stearns, L. Tan, and N. Magotra. A Technique for Lossless Compression of Seismic Data. In *Proceedings IGARSS '92*, pages 681-683. IEEE, 1992.

[22] I. Asanuma, Y. Kuroda, K. Yoshifumi, G. Kai, K. Muneyama, and S.-I. Isshii. Development of Data Compression and Transfer System. In *Oceans '89. Part 3: Navigation; Remote Sensing; Underwater Vehicles/Exploration.* IEEE, 1989.

[23] M.P. Cough, R. German, J.A. Thompson, and L.J.C. Wooliscroft. Simulation of Satellite Geophysical Data. *International Journal of Remote Sensing*, 8:1219-1227, 1987.

[24] B.R. Epstein, R. Hingorami, J.M. Shapiro, and M. Czigler. Multispectral Image Compression by Wavelet/Karhunen-Loeve Transformation. In *Proceedings IGARSS '92*, pages 672-674, 1992.

[25] C.Y. Chang, R. Kwok, and J.C. Curlander. Spatial Compression of Seasat SAR Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, pages 673-685, September 1988.

[26] J. Max. Quantizing for Minimum Distortion. *IRE Transactions on Information Theory*, IT-6:7-12, March 1960.

[27] S.P. Lloyd. Least Squared Quantization in PCM. *IEEE Transactions on Information Theory*, IT-28:129-137, March 1982.

[28] R. Kwok and W.T. Johnson. Block Adaptive Quantization of Magellan SAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, pages 375-383, July 1989.

[29] P.C. Dubois and L. Norikane. Data Volume Reduction for Imaging Radar Polarimetry. In *IGARSS '87*, pages 691-696, May 1987.

[30] K. Sayood and K. Anderson. A differential lossless image compression algorithm. *IEEE Transactions on Signal Processing*, 40(1):236-241, January 1992.

[31] M.C. Rost and K. Sayood. An Edge Preserving Differential Image Coding Scheme. *IEEE Transactions on Image Processing*, 1:250-256, April 1992.

[32] M.J. Shalkhauser and W.A. Whyte Jr. Digital CODEC for Real Time Signal Processing at 1.8 bpp. In *Global Telecommunication Conference*, 1989.

[33] K. Sayood and J.C. Borkenhagen. Use of Residual Redundancy in the Design of Joint Source/Channel Coders. *IEEE Transactions on Communications*, 39:838-846, March 1991.

[34] Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantization design. *IEEE Transactions on Communications*, COM-28:84-95, Jan. 1980.

[35] H. Abut, editor. *Vector Quantization*. IEEE Press, 1990.

[36] Smita Gupta and Allen Gersho. Feature Predictive Vector Quantization of Multispectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 30:491-501, May 1992.

[37] D.D. Giusto. On the Compression of Multispectral Images. In *Proceedings IGARSS '90*, pages 1651-1654. IEEE, 1990.

[38] J.D. Gibson and K. Sayood. Lattice Vector Quantization. In *Advances in Electronics and Electron Physics*, volume 72, pages 259-330. Academic Press, 1988.

[39] K. Sayood, J.D. Gibson, and M.C. Rost. An Algorithm for Uniform Vector Quantizer Design. *IEEE Transactions on Information Theory*, IT-30:805-814, November 1984.

[40] T.R. Fischer. A Pyramid Vector Quantizer. *IEEE Transactions on Information Theory*, IT-32:568-583, July 1986.

[41] G.K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):31-44, April 1991.

[42] K.R. Rao and P. Yip. *Discrete Cosine Transform - Algorithms, Advantages, Applications*. Academic Press, 1990.

[43] V. Cappelini and E. Del Re. Some Efficient Methods for Image Data Compression. In A. Luque, A.R. Figueiras Vidal, and J.M.R. Delgado, editors, *Proceedings of MELECON '85*, pages 473-475. IEEE, 1985.

[44] P. Ruffer, F. Rabe, and F. Gliem. A DCT Image Data Processor for use on the Huygens Titan Probe. In *Proceedings IGARSS '92*, pages 678-680. IEEE, 1992.

[45] H. Michalik, P. Ruffer, and F. Gliem. Versatile Online Image Data Compression for Spaceborne Applications. In *Proceedings IGARSS '91*, pages 1443-1447. IEEE, 1991.

[46] C. Mailhes, P. Vermande, and F. Castanie. Spectral Image Compression. *Journal of Optics*, 21:121-132, 1990.

Biography

Khalid Sayood (S'78-M'79-S'80-M'82) was born in Pakistan in 1956. He received his undergraduate education at the Middle East Technical University, Ankara, Turkey, and the University of Rochester, Rochester, NY. He received the B.S. and M.S. degrees from the University of Rochester, and the Ph.D. degree from Texas A&M University College Station, TX, in 1977, 1979, and 1982 respectively, all in electrical engineering.

He joined the Department of Electrical Engineering at the University of Nebraska-Lincoln, in 1982, where he is currently serving as a Professor. His current research interests include data compression, joint source/channel coding, communication networks, and biomedical applications.

Dr. Sayood is a member of Eta Kappa Nu and Sigma Xi.

Compression of Color-mapped Images*

A.C. Hadenfeldt
University of Nebraska Medical Center
Omaha, Nebraska 68198

and

K. Sayood
Department of Electrical Engineering
and
Center for Communication and Information Science
University of Nebraska-Lincoln
Lincoln, NE 68588
Telephone: (402) 472-6688
FAX: (402) 472-4732
email: ksayood@eecomm.unl.edu

COMSOC Technical Committee: Signal Processing and Communication Electronics
Hot topic session number: HT28

Abstract

In a standard image coding scenario, pixel-to-pixel correlation nearly always exists in the data, especially if the image is a natural scene. This correlation is what allows predictive coding schemes (e.g., DPCM) to perform efficient compression. In a color-mapped image, the values stored in the pixel array are no longer directly related to the pixel intensity. Two color indices which are numerically adjacent (close) may point to two very different colors. The correlation still exists, but only via the colormap. This fact can be exploited by sorting the color map to reintroduce the structure. In this paper we study the sorting of colormaps and show how the resulting structure can be used in both lossless and lossy compression of images.

* This work was supported by the NASA Goddard Space Flight Center (NAG 5-1612) and the NASA Lewis Research Center (NAG 3-806).

1 Introduction

Many lower-cost image display systems use color-mapped (or pseudo-color) displays. While there has been considerable attention devoted to the compression of monochrome and full-color images, the compression of color-mapped images has not received similar attention.

The human eye can distinguish hundreds of thousands of different colors in a color space, depending on viewing conditions [1]. A full-color (also called true-color) frame buffer provides a means of displaying this wide range. Such a system is illustrated in Figure 1.

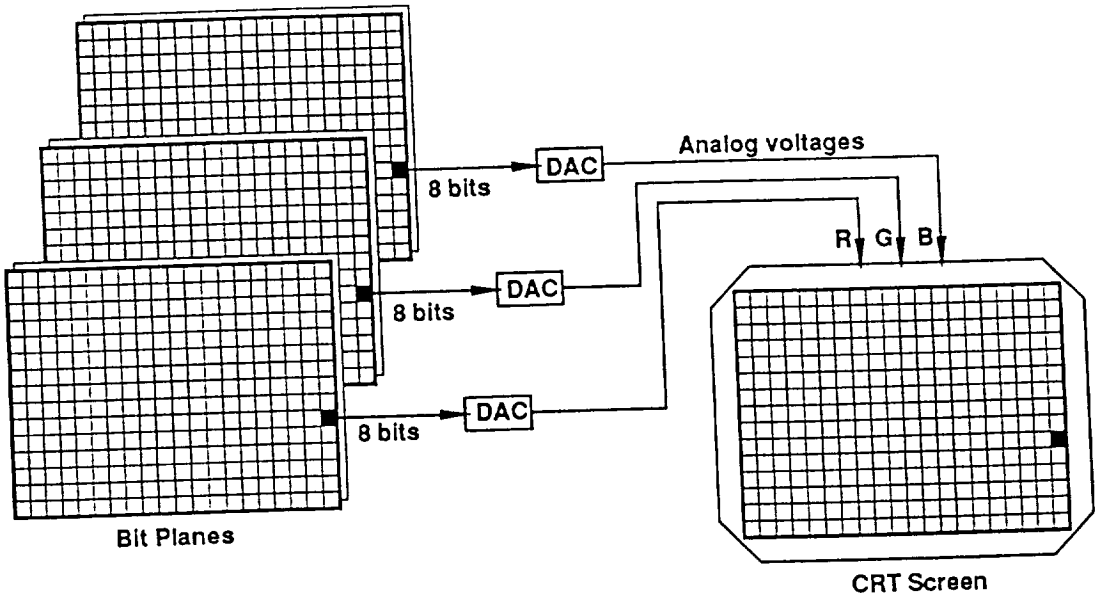


Figure 1 Full-Color Frame Buffer

Many applications of digital images benefit from, or require, color capabilities to be effective. If a full-color display is used, an application may become too costly to implement practically. Also, the images involved require large amounts of storage space, whether in display memory or on a mass-storage device. A less expensive solution is needed.

These applications naturally lead to the pseudo-color or color-mapped frame buffer, shown in Figure 2. This type of display is typical of those found on personal computers and workstations. A smaller amount of image memory is required, one-third that of the full-color system example. The values stored in memory are used as indices into a 24-bit table, the colormap.

Each entry in the colormap consists of 8-bit values for the red, green, and blue portions of the pixel. These three values are then passed through DACs to the red, green, and blue electron guns of the CRT, as with full-color system. The color-mapped system allows the display of a small number of colors at a time, 2^8 for the system shown in the figure, which can be selected from a larger set of colors (2^{24} for this example). By careful selection of the colors in the colormap, a large variety of images can be displayed, often with quality approaching that of a full-color display system.

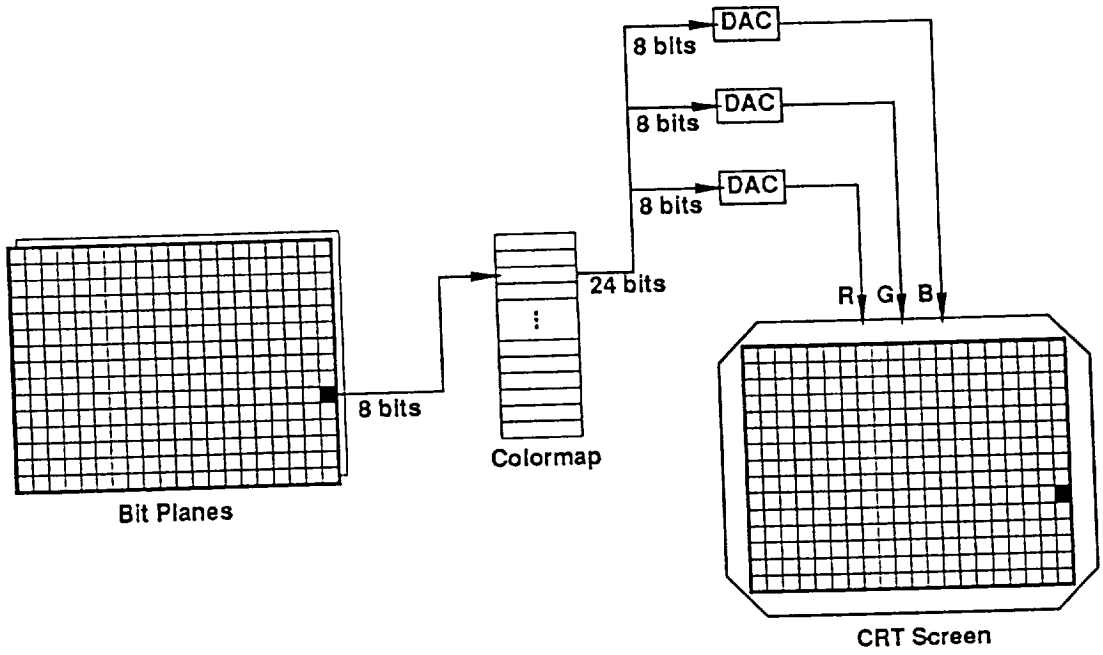


Figure 2 Color-Mapped Frame Buffer

The use of the colormap, however, disguises the spatial structure in the image. An indication of this can be obtained by calculating the zero-th and first order entropies of the image. These quantities were computed using the index arrays for the four test images shown in Figure 3, and are listed in Table 1.

Table 1 Entropies of the Source Images

Image	H_0	H_1
Lena	7.617	7.413
Park	7.470	7.797
Omaha	7.242	7.165
Lincoln	5.916	6.674

The large values of H_1 in the table verify that the spatial correlation in the image pixel values has been reduced by the color-mapping process. The values of H_0 are also relatively large, a direct result of selecting an 8-bit colormap. The data compression due to the color quantization process implies that the color index values stored in the image are more critical than similar values in, for example, an achromatic image. To further verify this, an experiment was conducted in which errors were introduced in the least significant bit of a color-mapped image, similar to what might be encountered if the color index values were quantized. The resultant images were of poor subjective quality at best, and often completely unrecognizable. Since quantization is a part of many popular source coding schemes, the available choices for compression schemes become limited.

2 Colormap Sorting

In the previous section, several problems unique to color-mapped images were discussed. The root of these problems is that the colormap indices stored in the image have little relationship with each other, which complicates coding for progressive transmission. In this section, methods of restoring this relationship are discussed.

Colormap sorting is a combinatorial optimization problem. Treating the K colormap entries as vectors, the problem is defined as follows. Given a set of vectors $\{a_1, a_2, ..., a_K\}$ in a three-dimensional vector space and a distance measure $d(i,j)$ defined between any two vectors a_i and a_j, find an ordering function $L(k)$ which minimizes the total distance D:

$$D = \sum_{k=1}^{K-1} d(L(k),\ L(k+1)) \tag{1}$$

The ordering function L is constrained to be a permutation of the sequence of integers $\{1, ..., K\}$. Another possibility results when the list of colormap entries is considered as a ring structure. That is, the colormap entry specified by $L(K)$ is now considered to be adjacent to the entry specified by $L(1)$. In this case, an additional term of $d(L(K), L(1))$ is added to the distance formula D.

The sorting problem is similar to the well-known travelling salesman problem, and is identical if the colormap is considered as a ring structure. As such, the problem is known to be NP-complete [3], and the number of possible orderings to consider is $1/2[(K-1)!]$[4]. Algorithms exist which can solve the problem exactly [4][5]; however, these algorithms are computationally feasible only for K no greater than about 20. Efficient algorithms for locating a local minimum exist [4] for $K \leq 145$. For large colormaps such as $K = 256$, another approach is necessary. Two techniques were tested. The first is a "greedy" technique, discussed in Section 2.1. The second is an algorithm which has performed well in practice, known as *simulated annealing*. Simulated annealing was chosen as the sorting method for the colormaps in this work, and is described in more detail in Section 2.2.

The distance metric d was chosen to be (unweighted) Euclidean distance, and different color spaces were investigated. Three color spaces were selected: the NTSC RGB space, the CIE L*a*b* space, and the CIE L*u*v* space. The NTSC RGB space was chosen since it corresponds to the color primaries of the original images. Color spaces which can be linearly transformed to the NTSC RGB space were not considered, since the use of an unweighted Euclidean distance measure would give similar results for such a color space. The two CIE color spaces were selected since they provide a means to measure perceptual color differences.

2.1 Sorting Using Simulated Annealing

Simulated annealing [3][6] is a stochastic technique for combinatorial minimization. The basis for the technique comes from thermodynamics and observations concerning the properties of materials as they are cooled. The technique described in this section is based on the implementation in [6].

To illustrate this concept, consider an iron block. At high temperatures, the iron molecules move freely with respect to each other. If the block is *quenched* (cooled very quickly), the molecules will be locked together in a high-energy state. On the other hand, if the block is *annealed* (cooled very slowly), the molecules will tend to redistribute themselves as they lose energy, with the result being a lower energy lattice which is much stronger. The distribution of molecular energies is characterized by the Boltzmann distribution:

$$P(E) \sim e^{-E/kT} \tag{2}$$

where E is the energy state, T is the temperature, and k is Boltzmann's constant. The significance of this distribution is that even at low temperatures, there is some probability that a molecule will have a high energy. In a combinatorial optimization situation, the Boltzmann distribution can be used to temporarily allow increases in the cost function, while still generally striving to achieve a minimum.

Solving the colormap sorting problem involves selecting each color only once while minimizing the sum of the distances between the colors. To find a solution using simulated annealing, an initial path through the nodes (colors) is chosen, and its cost computed. The algorithm then proceeds as follows:

1. Select an initial temperature T and a cooling factor α.
2. Choose a temporary new path by perturbing the current path (see below), and compute the change in path cost, $\Delta E = E_{\text{new}} - E_{\text{old}}$. If $\Delta E \leq 0$, accept the new path.
3. If $\Delta E > 0$, randomly decide whether or not to accept the path. Generate a random number r from a uniform distribution in the range [0, 1), and accept the new path if $r < \exp(-\Delta E/T)$.
4. Continue to perturb the path at the current temperature for I iterations. Then, "cool" the system by the cooling factor: $T_{\text{new}} = \alpha T_{\text{old}}$. Continue iterating using the new temperature.
5. Terminate the algorithm when no path changes are accepted at a particular temperature.

The decision-making process is known as the *Metropolis algorithm*. Note that the decision process will allow some changes to the path which increase its cost. This makes it possible for the simulated annealing method to avoid easily being trapped in a local minimum of the cost function. Hence, the algorithm is less sensitive to the initial path choice.

For the images of this work, initial values of T ranged from 80 to 500, depending on the color space used. The cooling factor α was usually chosen as 0.9. The simulated annealing algorithm seemed to be most sensitive to the choice of this value, as values outside the range [0.85, 0.95] caused the cooling to occur too slowly or too quickly. The number of iterations per temperature I was chosen as 100 times the number of nodes (colors), or 25,600. However, to improve the execution speed of the algorithm an improvement suggested by [6] was added, which causes the algorithm to proceed to the next temperature if (10)(number of nodes) = 2560 successful path changes are made at a given temperature.

Also, a method for perturbing the path must be selected. In this work, the perturbations were made using the suggestions of Lin [4][6]. At each iteration, one of two possible changes to the path are made, chosen at random. The first is a *path transport*, which removes a segment of the current path and reinserts it at another point in the path. The location of the segment, its length, and the new insertion point are chosen at random. The second perturbation method, called *path reversal*, removes a segment of the current path and reinserts it at the same point in the path, but with the nodes in reverse order. The location and length of the segment are again randomly chosen.

The algorithm outlined in the previous paragraphs formulates colormap sorting as a travelling salesman problem. This type of problem usually assumes a complete tour will be made (i.e., the salesman desires to return to the original city). Hence, the colormap is assumed to have a ring-like structure. However, the simulated annealing technique can also be used if this is not the case, allowing the colormap to be considered as a linear list structure. Experiments using both structures were conducted.

3 Colormap Sorting and Lossless Compression

The results of sorting the colormaps of the test images using simulated annealing are shown in the following tables. Table 2 shows results for sorting the colormap as a circular ring structure, while Table 3 shows the results of sorting the colormap as a linear structure. Given in the tables are values for the resulting first-order entropy and the final path cost (the distance measure D).

Table 2 Resultant Images With Circularly Sorted Colormaps

Image Name	RGB Space		L*a*b Space		L*u*v* Space	
	Cost	H_1	Cost	H_1	Cost	H_1
Lena	13.88	5.641	857.80	5.627	208.49	5.480
Park	19.32	6.325	1609.46	6.330	310.41	6.218
Omaha	11.04	6.209	1081.82	6.303	363.21	6.178
Lincoln	10.62	5.513	1193.88	5.831	224.06	5.478

Table 3 Resultant Images With Linearly Sorted Colormaps

Image Name	RGB Space		L*a*b Space		L*u*v* Space	
	Cost	H_1	Cost	H_1	Cost	H_1
Lena	11.68	5.575	847.29	5.933	200.31	5.512
Park	15.66	6.260	1509.25	6.775	292.29	6.546
Omaha	10.81	6.532	1004.69	6.554	283.66	6.199
Lincoln	10.61	5.774	1177.80	6.120	204.64	5.735

Note that the zero-order entropy H_0 is not changed by the sorting process, since permuting the colormap entries does not change the frequency of occurrence of a particular color. The lower first-order entropies of the resultant images indicate that some of the spatial correlation between color indices has been restored in each case. The sorting results for the NTSC RGB space show that sorting in this space yields good results, if entropy reduction (the first goal stated above) is the goal. However, the L*u*v* space sorting gives better results, with the added advantage that the perceptual differences between colormap entries has been considered. Hence, the resultant images from this sort should also be able to accept quantization errors while maintaining good subjective quality, the second goal stated previously. We examine this further in the next section. In terms of lossless compression, the sorting has resulted in a drop of 2 bits per pixel for the

Lena image and 1 to 1.5 bits per pixel for the other images. For a 512x512 image this translates to a savings of between 32,768 to 65,536 bytes per image. For a large database of images this could be a considerable saving.

4 Colormap Sorting and Lossy Compression

The sorting of the colormap restores some perceptual structure to the colormap indices in the sense that indices close in numerical value are also close in some perceptual sense. Therefore it should be possible to introduce errors into the indices without destroying the image. To verify this hypothesis, we dropped the three least significant bits of the L*u*v*-sorted Park images. Good subjective results were obtained using quantization levels down to as low as 5 bits/pixel from the 8-bit original. Figure 4 shows the colormap for the Park image, before and after sorting. The sorted colormap shown was sorted as a linear list in L*u*v* space. Figure 5 shows the result of quantizing the Park image to 5 bits/pixel, before and after the colormap has been sorted. A caveat is in order here. While the distance between the eight-bit indices have more perceptual meaning, the sorted colormap image should not be assumed to have the same properties as an eight-bit monochrome image. In some cases, if the distance between the original and reconstructed (compressed and decompressed) indices is large enough, there might be a drastic change in color between those pixels in the original and reconstructed image. In the monochrome case large distances would correspond to changes in shading which might be overlooked by the viewer.

To see how well the sorted color-mapped images lend themselves to lossy compression we compress them using particular implementations of two popular lossy compression techniques, the Discrete Cosine Transform (DCT) and Differential Pulse Code Modulation (DPCM).

4.1 DCT Coding of Color-mapped Images

In Figure 6 we coded the Lena image with the unsorted colormap at two bits per pixel using the unsorted color map. As can be seen from the figure, the original image is totally lost and all that remains is seemingly random colors. It should be noted that for eight-bit monochrome images,

DCT coding at two bits per pixel generally provides a reconstruction which is indistinguishable from the original.

In Figure 7 we show the same image, this time with the sorted color map, coded at two bits per pixel with the fixed bit allocation. [7] The images in Figure 8 were coded at two and one bit per pixel using the JPEG [8] algorithm.[1] Note that while the image coded using the fixed bit allocation shown in Figure 7 is far superior to the image in Figure 6 there are still quite a few annoying artifacts. This is because of the nonadaptive nature of the algorithm which, while it minimizes the *average* error, may permit the introduction of large errors in individual blocks. As the color-mapped images are particularly sensitive to large errors, this could account for the low quality reproduction. The JPEG algorithm adapts its bit allocation on a block-by-block basis. Therefore, the image in Figure 8(b) which is coded at half the rate of the image in Figure 7 still provides superior quality.

4.2 DPCM Coding of Color-mapped Images

Standard DPCM coding of color-mapped images is problematic because in the busy regions of images, especially edges, the prediction error is generally large, leading to large overload noise values. In monochrome images these noise values result in a blurred look around edges, which may be acceptable for certain application. However, in color-mapped images these noise values will result in splotches of different colors. The Edge Preserving DPCM (EPDPCM) system avoids this problem by the use of a recursively indexed quantizer [9,10], in which the magnitude of the quantization error is always bounded by $\frac{\Delta}{2}$. This attribute makes it ideal for application to the coding of color-mapped images. Another advantage of the EPDPCM system is that, as the quantizer output alphabet can be kept small without incurring overload error, the output is amenable to entropy coding.

Results using the EPDPCM system are shown in Figure 10. The image in Figure 10(a) was coded at a rate of 2 bits per pixel, while the image in Figure 10(b) was coded with 1.35 bpp.

---

1 The JPEG coded images were coded using software from the independent JPEG foundation.

The advantage of DPCM systems over transform coding systems is their low complexity and higher speed. However, the reconstruction quality obtained using transform coding systems is generally significantly higher than that of DPCM systems at a given rate. Comparing Figure 10(a) and 8(a), this is obviously not the case for the sorted colormapped images. In fact, the quality of the two-bit EPDPCM coded image is actually somewhat higher than the two-bit DCT coded image. Thus using the EPDPCM system provides advantages both in terms of complexity and speed, and reconstruction quality.

5 Conclusion

In this paper we have shown that use of sorted colormaps makes color-mapped images amenable to both lossless and lossy compression. For lossy compression conventional wisdom dictates the use of DCT coding for most types of images. However, for color-mapped images DPCM coding might be more advantageous.

6 References

[1] Foley, J.D., van Dam, A., S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice (Second Edition)*, Reading, MA: Addison-Wesley, 1990.

[2] *Graphics Interchange Format (GIF) Specification*, CompuServe, Inc., Columbus, OH, June 1987.

[3] Aarts, E., and J. Korst, *Simulated Annealing and Boltzmann Machines*, New York: John Wiley and Sons, 1989.

[4] Lin, S., "Computer Solutions of the Traveling Salesman Problem," *Bell System Technical Journal*, pp. 2245–2269, December 1965.

[5] Bellman, R.E., and S.E. Dreyfus, *Applied Dynamic Programming*, Princeton, NJ: Princeton University Press, 1962.

[6] Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, New York: Cambridge University Press, 1988.

[7] Jayant, N.S. and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.

[8] Wallace, G.K., "The JPEG still picture compression standard," *Communications of the ACM*, 34(4):31–44, April 1991.

[9] Rost, M.C. and K. Sayood, "An Edge Preserving Differential Image Coding Scheme," *IEEE Transactions on Image Processing*, 1:250–256, April 1992.

[10] Sayood, K. and S. Na. "Recursively Indexed Quantization of Memoryless Sources," *IEEE Transactions on Information Theory*, IT-38, November 1992.

List of Figures

Figure 3. Test Images

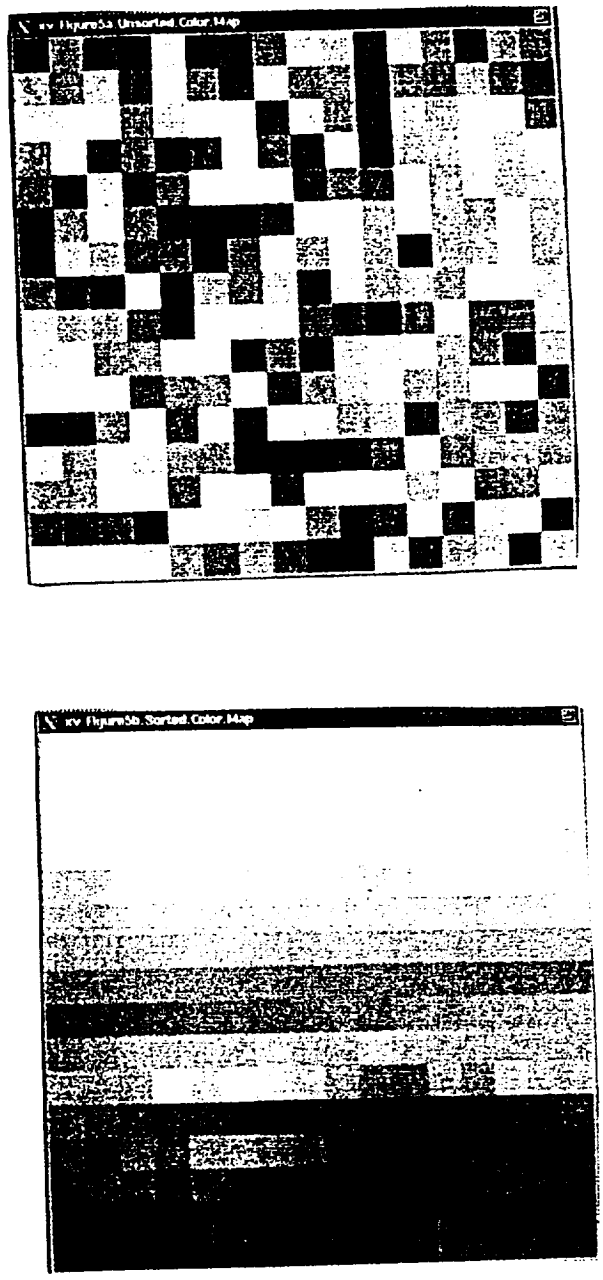


Figure 4. Colormap for Park Image (a) before and (b) after sorting

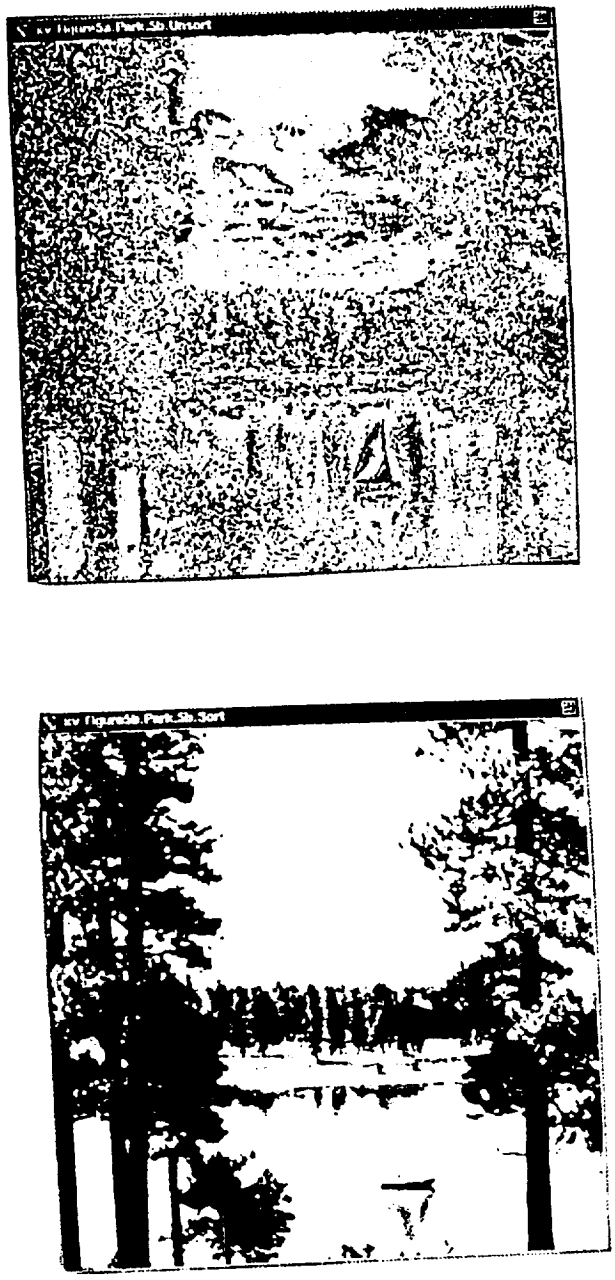


Figure 5. Park image quanitzed to five bits per pixel with (a) unsorted and (b) sorted colormaps

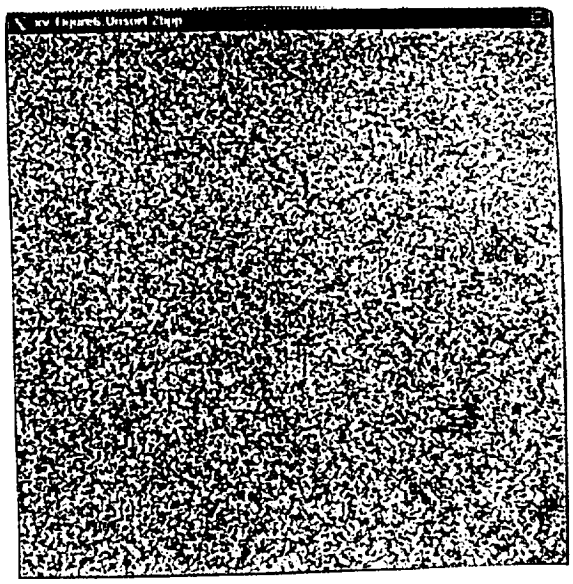


Figure 6. Lena image with unsorted colormap coded at two bits per pixel using JPEG DCT algorithm

A Robust Coding Scheme for Packet Video

Yun-Chung Chen, Khalid Sayood, *Member, IEEE,* and Don J. Nelson, *Senior Member, IEEE*

Abstract—**We present a layered packet video coding algorithm based on a progressive transmission scheme. The algorithm provides good compression and can handle significant packet loss with graceful degradation in the reconstruction sequence. Simulation results for various conditions are presented.**

I. Introduction

DUE to the rapid evolution in the fields of image processing and networking, video information will be an important part of tomorrow's telecommunication system. Up to now, video transmission has been mainly transported over circuit-switched networks. It is quite likely that packet-switched networks will dominate the communications world in the near future. Asynchronous transfer mode (ATM) techniques in broadband-ISDN can provide a flexible, independent and high-performance environment for video communication. Therefore, it is necessary to develop techniques for video transmission over such networks.

The classic approach in circuit switching is to provide a "dedicated path," thus reserving a continuous bandwidth capacity in advance. Any unused bandwidth capacity on the allocated circuit is therefore wasted. Rapidly varying signals, like video signals, require too much bandwidth to be accommodated by a standard circuit-switching channel. With a certain amount of capacity assigned to a given source, if the output rate of that source is larger than the channel capacity, quality will be degraded. If the generating rate is less than the available capacity, the excess channel capacity is wasted. The use of packet networks allows for the utilization of channel sharing protocols between independent sources and can improve channel utilization. Another point that strongly favors packet-switched networks is the possibility that the integration of services in a network will be facilitated if all of the signals are separated into packets with the same format.

Some coding schemes which support packet video have been explored. Verbiest and Pinnoo proposed a DPCM-based system which is comprised of an intrafield/interframe predictor, a nonlinear quantizer, and a variable length coder [1]. Their codec obtains stable picture quality by switching between three different coding modes: intrafield DPCM, interframe DPCM, and no replenishment. Ghanbari has simulated a two-layer conditional replenishment codec with a first layer based on hybrid DCT-DPCM and second layer using DPCM [2]. This scheme generates two type of packets: "guaranteed packets" contain vital information and "enhancement packets" contain "add-on" information. Darragh and Baker presented a sub-band codec which attains a user-prescribed fidelity by allowing the encoder's compression rate to vary [3]. The codec's design is based on an algorithm that allocates distortion among the sub-bands to minimize channel entropy. Kishino *et al.* describe a layered coding technique using discrete cosine transform coding, which is suitable for packet loss compensation [4]. Karlsson and Vertterli presented a sub-band coder using DPCM with a nonuniform quantizer followed by run-length coding for baseband and PCM with run-length coding for nonbaseband [5]. In this paper, a different coding scheme based on a progressive transmission scheme called Mixture Block Coding with Progressive Transmission (MBCPT) [6], [7] is investigated. Unlike the methods mentioned above, MBCPT does not use decimation and interpolation filters to separate the signals into sub-bands. However, it does have the attractive property of dealing separately with high frequency and low frequency information. This separation is obtained by the use of variable blocksize transform coding.

This paper is organized as follows. First, some of the important characteristics and requirements of packet video are discussed. In Section III, the coding scheme called mixture block coding with progressive transmission (MBCPT) is presented. In Section IV, a network simulator used in testing the scheme is introduced. In Section V the simulation results are discussed. Finally, in Section VI the paper is summarized.

II. Characteristics of Packet Video

The demand for various services, such as telemetry, terminal and computer connections, voice communications, and full-motion high-resolution video, along with the wide range of bit rates and holding times they represent, provides an impetus for building a Broadband Integrated Service Digital Network (B-ISDN). B-ISDN is a projected worldwide public telecommunications network that will service a wide range of user needs. The continuing advances in the technology of optical fiber transmission and integrated circuit fabrication have been driving forces to realize B-ISDN. The idea of B-ISDN is to build a complete end-to-end switched digital telecommunication network with broadband channels. Still to be precisely defined by CCITT, with fiber transmission, H4 has an access rate of about 135 Mbps.

Packet-switched networks have the unique characteristics of dynamic bandwidth allocation for transmission and switching resources, and the elimination of channel structure. They acquire and release bandwidth as needed. Because the video signals vary greatly in bandwidth requirement, it is attractive

Paper approved by the Editor for Communications Systems of the IEEE Communications Society. Manuscript received August 29, 1990; revised March 29, 1991. This work was supported by the NASA Goddard Space Flight Center under Grant NAG5-916.

The authors are with the Department of Electrical Engineering, Center for Communication and Information Science, University of Nebraska-Lincoln, Lincoln, NE, 68588.

IEEE Log Number 9201526.

to utilize a packet-switched network for video coded signals. Allowing the transmission rate to vary, video coding based on packet transmission permits the possibility of keeping the picture quality constant, by implementing "bandwidth on demand." There are three main merits when transmitting video packets over a packet-switched network.

1) Improved and consistent image quality: If video signals are transmitted over fixed-rate circuits, there is a need to keep the coded bit rate constant, resulting in image degradation accompanying rapid motion.

2) Multimedia integration: As mentioned above, integrated broadband services can be provided using unified protocols.

3) Improved transmission efficiency: Using variable bit-rate coding and channel sharing among multiple video sources, scenes can be transmitted without distortion if other sources, at the same time, are without rapid motion.

However video transmission over packet networks also has the following drawbacks.

1) The time taken to transmit a packet of data may change from time to time.

2) Packets may be delayed to the point where, because of constraints due to the human visual system, they have to be discarded.

3) Headers of packets may be changed because of errors and delivered to the wrong receiver.

It has to be emphasized that the delay/lost effect can reach very high levels if the combined users' requirements exceeds the acquirable bandwidth and may seriously damage the quality of the image.

When the signals transmitted in the network are nonstationary and circuit-switching is used with limited bandwidth, a buffer between the coder and the channel is needed to smooth out the varying rate. If the amount of data in the buffer exceeds a certain threshold, the encoder is instructed to switch into a coding mode that has lower rate but worse quality to avoid buffer overflow. In packet-switched networks, asynchronous time division multiplexing (ATDM) can efficiently absorb temporal variations of the bit-rate of individual sources by smoothing out the aggregate of several independent streams in the common network buffers [8].

To deliver packets in a limited time and provide a real time service is a difficult resource allocation and control problem, especially when the source generates a high and greatly varying rate. In packet-switched networks, packet losses are inevitable, but use of a packet-switched network yields a better utilization of channel capacity. However, it should be noted that the varying rate requirements of the video coder may not be synchronized with the variations in available channel capacity which changes depending on the traffic in the network. Therefore, the interactions between the coder and the network have to be considered and incorporated into the requirements for the coder. These requirements include the following.

1) Adaptability of the coding scheme: The video source we are dealing with has a varying information rate. So it is expected that the encoder should generate different bit rates by removing the redundancy. When the video is still, there is no need to transmit anything.

2) Insensitivity to error: The coding scheme has to be robust to the packet loss so that the quality of the image is never seriously damaged. Remember that retransmission is impossible because of the tight timing requirement.

3) Resynchronization of the video: Because of the varying packet-generating rate and the lack of a common clock between the coder and the decoder, we have to find a way to reconstruct the received data which is synchronous to the display terminal.

4) Control coding rate: Sensing the heavy traffic in the network, the coding scheme is required to adjust the coding rate by itself. In the case of a congested network, the coder could be switched to another mode which generates fewer bits with a minimal degradation of image quality.

5) Parallel architecture: The coder should preferably be implemented in parallel. That allows the coding procedure to be run at a lower rate in many parallel streams.

In the next section, we investigate a coding scheme to see how well it satisfies the above requirements.

III. Mixture Block Coding with Progressive Transmission

Mixture block coding (MBC) is a variable-blocksize transform coding algorithm which codes the image with different blocksizes depending upon the complexity of that block area. Low-complexity areas are coded with a large blocksize transform coder while high-complexity regions are coded with small blocksize. The complexity of the specific block is determined by the distortion between the coded and original image when the same number of bits are used to code each block. A more complex image block has higher distortion. The advantage of using MBC is that it does not process different complex regions with the same blocksize. That means MBC has the ability to choose a finer or coarser coding scheme to deal with different complex parts of the same image. With the same rate, MBC is able to provide an image of higher quality than a coding scheme which codes different complex regions with the same blocksize coder.

When using MBC, the image is divided into maximum blocksize blocks. After coding, the distortion between the reconstructed block and the original block is calculated. The block being processed is subdivided into smaller blocks if that distortion fails to meet the predetermined threshold. The coding-testing procedure continues until the distortion is small enough or the smallest blocksize is reached. In this scheme, every block is coded until the reconstructed image is satisfactory and then moves to the next block.

Mixture block coding with progressive transmission (MBCPT) is a coding scheme which combines MBC and progressive coding. Progressive coding is an approach that allows an initial image to be transmitted at a lower bit rate which can later be updated [9]. In this way, successive approximations converge to the target image with the first approximation carrying the "most" information and the following approximations enhancing it. The process is like focusing a lens, where the entire image is transformed from low-quality into high-quality. In progressive coding, every

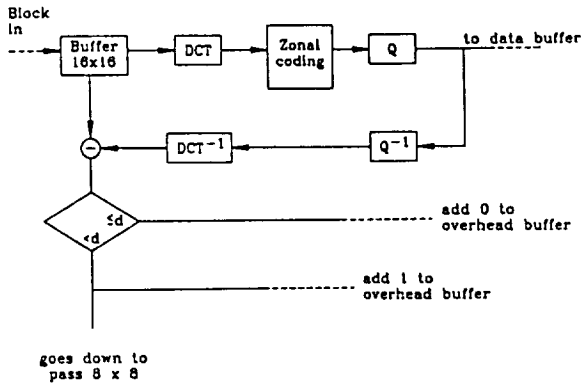


Fig. 1. Structure of the first pass consisting of 16 × 16 blocks for MBCPT.

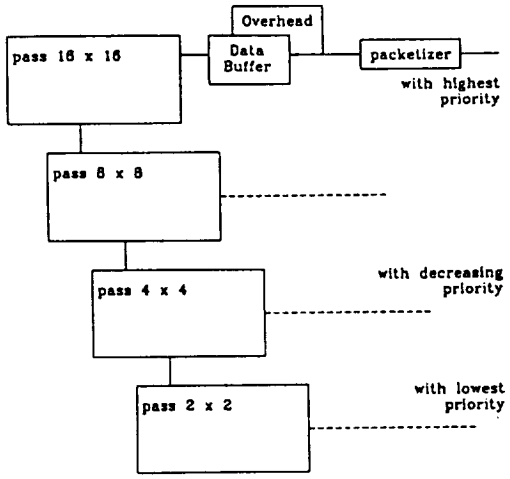


Fig. 2. Parallel structure for MBCPT.

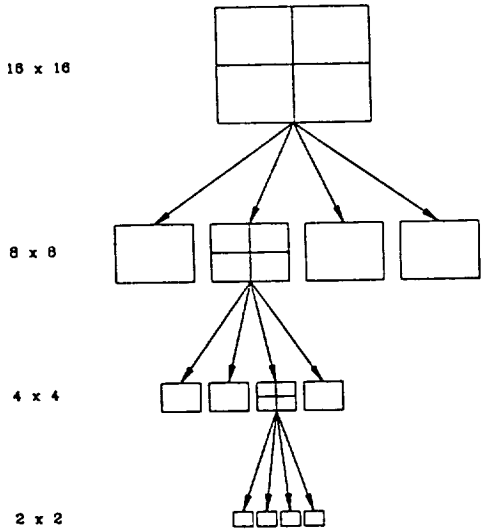


Fig. 3. Example of the quad tree structure.

pixel value, or the information contained in it, is possibly coded more than once and the total bit rate may increase due to different coding scheme and quality desired. Because only the gross features of an image are being coded and transmitted in the first pass, the processing time is greatly reduced for the first pass and a coarse version of the image can be displayed without significant delay. It has been shown that it is perceptually useful to get a crude image in a short time, rather than waiting a long time to get a clear complete image.

With different stopping criterion, progressive coding is suitable for dynamic channel capacity allocation. If a predetermined distortion threshold is met, processing is stopped and no more refining action is needed. The threshold value can be adjusted according to the traffic condition in the channel. Successive approximations (or iterations) are sent through the channel in progressive coding and lead the receiver to the desired image. If these successive approximations are marked with decreasing piority, then a sudden decrease in channel capacity may only cause the received image to suffer from quality degradation rather than total loss of parts of the images.

MBCPT is a multipass scheme in which each pass deals with different blocksizes. The first pass codes the image with maximum blocksize and transmits it immediately. Only those blocks which fail to meet the distortion threshold go down to the second pass which processes the difference image block (coming from the original and coded image obtained in the first pass) with smaller blocks. The difference image coding scheme continues until the final pass which deals with the minimum size block. At the receiving end, a crude image is obtained from the first pass in a short time and the data from following passes serve to enchance it. Fig. 1 shows the structure of a pass consisting of 16 × 16 blocks for MBCPT. Fig. 2 shows the parallel structure of MBCPT. Coding algorithms using quad trees have also been proposed by Dreizen [10] and Vaisey and Gersho [11]. In the quad tree coding structure of this paper, the 16 × 16 block is coded and the distortion of the block is calculated. If the distortion is greater than the predetermined threshold for 16 × 16 blocks, the block is divided into four 8 × 8 blocks for additional coding. This coding-checking procedure is continued until the only image blocks not meeting the threshold are those of size 2 × 2. Fig. 3 shows the algorithm.

The blocksize used in the coding scheme should be small enough for ease of processing and storage requirements, but large enough to limit the inter-block redundancy [12]. Large blocksizes result in higher compression, but it is very difficult to build real-time hardware for blocksizes larger than 16 × 16 because of the increase in the number of computations. So, 16 × 16 is chosen to be the largest blocksize. The minimum blocksize determines the finest visual qualtiy that is achievable in the busy area. If the minimum blocksize is too large, it is possible to observe the blockiness in the coded edge of spherical objects because the coding block is square. In order to match the zonal transform coding used in this paper, 2 × 2 is the smallest blocksize and there are four passes (16 × 16, 8 × 8, 4 × 4, 2 × 2) in this scheme. Figs. 4–7 shows images from the 4 passes.

After applying the discrete cosine transform, only four coefficients, including the dc and three lowest order frequency coefficients, are coded and the others are set to zero. The dc coefficient in the first pass is coded with an 8-bit uniform quantizer due to the fact that it closely reflects the average gray level for that image block and is hard to model. The dc coefficient in the subsequent passes follows a Laplacian

Fig. 4. Image reconstructed from first pass.

Fig. 5. Image reconstructed from first two passes.

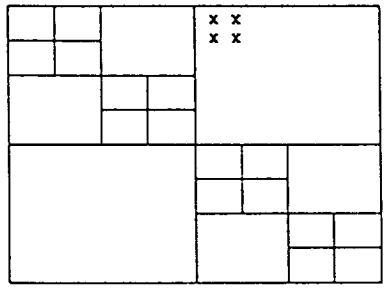
Fig. 6. Image reconstructed from first three passes.

model, and a 5-bit optimal Laplacian nonuniform quantizer is used to code it. The ac coefficients also follow a Laplacian model with a variance greater than that of the dc coefficient and can therefore also be coded using a Laplacian quantizer. As an alternative, an LBG vector quantizer with a 512 codebook size is used to quantize the vector which comprises the three ac coefficient. The initial threshold of each pass is selected beforehand and is readjustable during the operation according to the channel condition and quality required.

Because only partial blocks which fail to meet the distortion threshold need to be coded, side information is needed to instruct the receiver on how to reconstruct the image. One bit of overhead is needed for each block. If a block is to be divided, a 1 is assigned to be its overhead; if not, a 0 is assigned. The example shown in Fig. 8 has the following overhead: 1, 1001, 1001, 1001, 1001, 1001.

The interframe coder used in this paper is a differential scheme which is based on MBCPT. This coder processes

Fig. 7. Image reconstructed from four passes.



16 x 16

overhead = 1,1001,1001,1001,1001,1001

Fig. 8. Overhead assignment and zonal coding.

the difference image coming from the current frame and the previous frame which is locally decoded from the first three pass data. Fig. 9 shows the algorithm of this coder. Fig. 10 shows a different scheme which does the local decoding with all four passes. From Fig. 11, it can be seen that when there is no packet loss, the performances of these two schemes are quite the same. But when congestion occurs in the network, with the priorities assigned to packets, packets from pass 4 are expected to be discarded first. In this case, the performance (from Fig. 12) of the scheme in Fig. 9 is much better than the one in Fig. 10. Therefore the coding scheme in Fig. 9 is used in our simulation. In this paper, the Kronkite motion sequence from the USC database with 16 frames is used as the simulation source. Every image is 256 x 256 pixels with graylevels ranging from 0 to 255. It is similar to a video conferencing type image which has neither rapid motion nor scene changes. Due to this characteristic, advanced techniques like motion detection or motion compensation have not been used but could be implemented when broadcasting video.

From the datastream output that is listed in Table I, we can see that the data in pass 4 represents 30–40% of the entire data. This part of the data is involved in increasing the sharpness of the image and is usually labeled with the lowest priority in the network. We therefore call this the least significant pass (LSP). With a substantial possibility of being discarded due to low priority, those packets from pass 4 will not be used to reconstruct the locally decoded image and be stored in the frame memory. This prevents the packet loss error propagating into following frames if the lost packet belongs to pass 4.

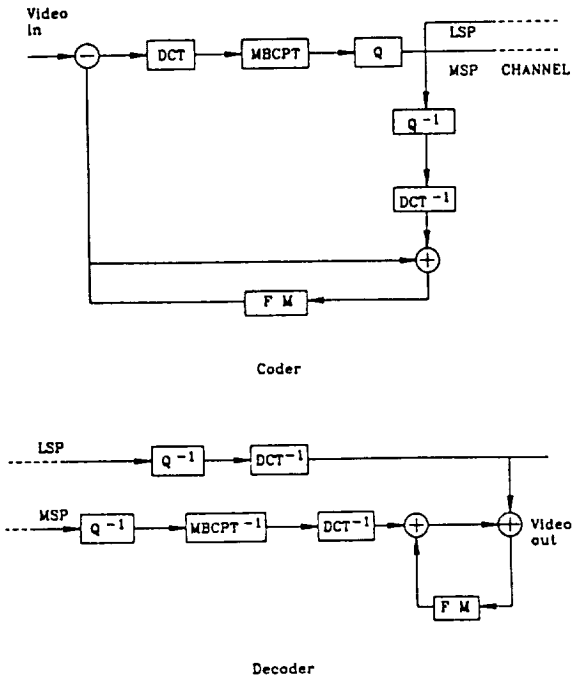


Fig. 9. Differential MBCPT coding scheme (1).

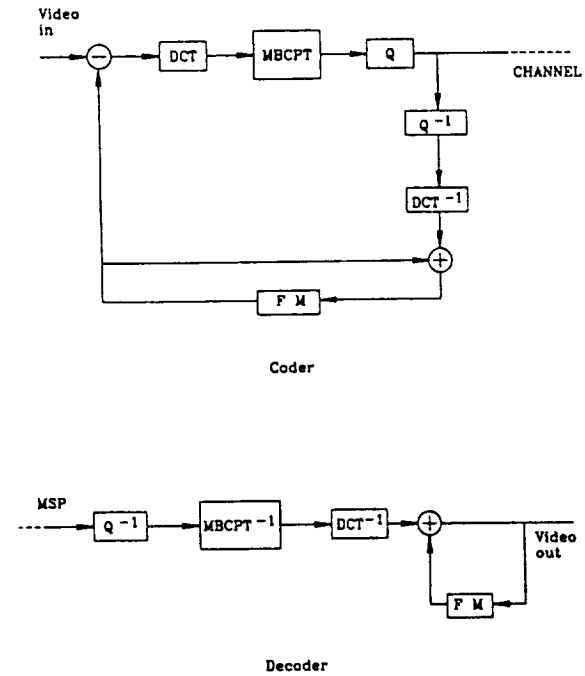


Fig. 10. Differential MBCPT coding scheme (2).

IV. Simulation Network

The network simulator used for this study was a modified version of an existing simulator developed by Nelson *et al.* [13]. A brief description of the simulator is provided here.

A. Introduction

As mentioned in Section II, tomorrow's integrated telecommunication network is a very complicated and dynamic structure. Its efficiency requires sophisticated monitoring and control algorithms with communication between nodes reflecting

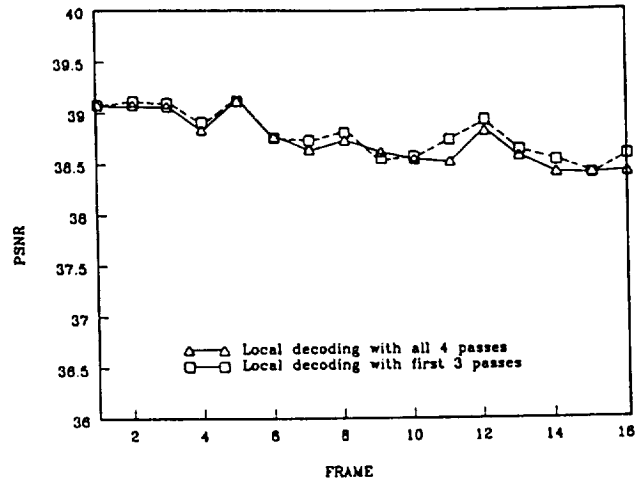


Fig. 11. Performance of two differential MBCPT schemes without packet loss.

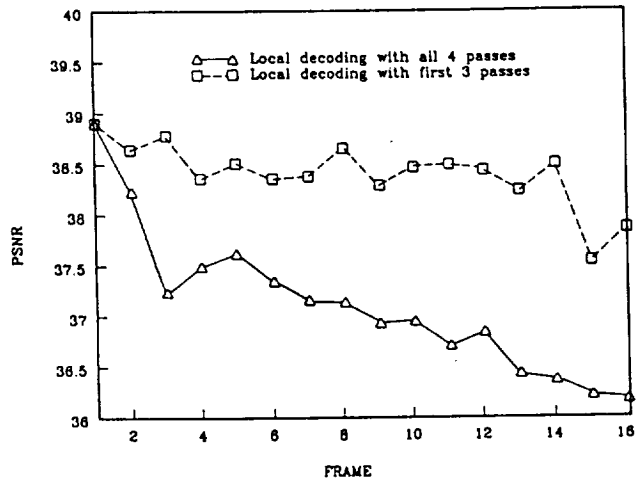


Fig. 12. Performance of the two MBCPT schemes with packet losses from pass 4.

the existing capacity and reliability of system components. The scheme for communicating information regarding the operating status is called the system protocol. Since the communication of system information must flow through the channel, it reduces the overall capacity of the physical layers, but hopefully provides a more efficient system overall. Therefore, system efficiency depends entirely upon these protocols, which, in turn, depend upon the system topology, communication channel properties, nodal memory and component reliability. Most network protocols have been developed to provide high reliability in topological structures with reasonably high channel reliability.

In order to fit into the purpose of this study, most modifications which were made to the simulator were in those modules concerning the network layer. Since the simulator is structured in modules which represent, to some degree, the ISO Model for packet switched networks, a more detailed description about the network layer modules follows.

B. The Network Layer and Basic Operation

The simulation of a layer at each node is represented by a "processor" and one or more "packet queues." All events are scheduled through the "Sim_Q" which drives the simu-

TABLE I
PERFORMANCE DEGRADATION DUE TO PACKET LOSS IN DIFFERENT PASSES

Frame #	PSNR with packet losses only in			
	Pass 4	Pass 3	Pass 2	Pass 1
0	40.30	40.30	40.30	40.30
1	40.59	40.37	40.12	37.55
2	40.07	39.02	36.15	31.99
3	39.70	38.19	35.82	31.70
4	40.19	38.35	36.31	30.18
5	39.65	38.05	35.21	28.35
6	38.74	36.27	33.23	26.07
7	38.59	35.58	31.52	24.61
8	38.68	34.96	30.81	23.27
9	38.51	34.33	29.85	21.77
10	39.48	34.31	29.86	21.54
11	39.26	34.01	29.67	21.90
12	38.83	33.75	29.57	22.00
13	38.54	33.09	29.46	22.30
14	38.86	33.21	29.52	22.34
15	39.47	33.24	29.37	22.33

lator. Initially, the processors are all idle, the packet queues are all empty and the only tasks scheduled are the arrival of messages at the various nodes. The simulator operation occurs by examining the next event and performing the task indicated. The task may result in the scheduling of additional events, generally referred to as task completion times. When a message or packet is placed in the input queue at a node for a given layer, the processor for that queue is marked as busy, the packet is removed from the queue, and the task to be performed by the processor is scheduled for completion. When the task is completed (as a result of the simulator reaching that point in time), the "processor" examines the queue. If the queue is empty, the processor is set idle; otherwise it removes the next message or packet from the queue and schedules the completion of the operation which must be preformed. The layers in the simulator are quite close in operation to the ISO transport, network and datalink layers.

1) The Session Layer: In the OSI model, the session layer (SL) allows users to establish "sessions" on local or remote systems. In the simulator, as mentioned above, it contains a relatively simple model of the subscribers, participates in flow-control, and acts as a statistics collector for messages arriving and delivered. At message arrival time (from Sim_Q), the session layer generates the "message" with all of its randomly selected attributes and if flow control or node hold-down are not in effect, submits it to the transport layer. It then schedules the next message arrival time. During initialization, the task "SL_Rcv_Msg" for each node is queued in Sim_Q for the arrival time of the first message at that node. When this task is executed by the simulator, a message packet is generated and placed in the transport queue. The arrival of the next message is then queued in Sim_Q with the same task and with an arrival time determined by the random number generator (Poisson Distributed). The only other task performed by the session layer is the "SL_Snd_Msg" task that simulates delivery of mesages to the subscribers, develops message statistics and "cleans up" the queues for messages delivered.

2) The Transport Layer: The basic function of the transport layer at the sending end is to receive the message from the session layer, place it in packets and pass the packets on to the network layer. At the receiving end, the packets are reassembled into a message for delivery to the session layer. To accomplish the complex task of assuring reliable delivery, there is a transport time-out mechanism at both the sending and receiving nodes and a message acknowledgement packet that is sent to the sending node when all packets for the message have been satisfactorily received. At the sending end, if a message acknowledgment is not received in the allotted time period, the message can be retransmitted. In the simulations reported in this paper, the retransmission feature was not used. At the receiving end, if all packets are not received in the specified period of time, the entire message is discarded. It is recognized that in some networks, packetization takes place at the network level, leaving the transport layer responsible only for message-level structures. Reassembly, depending upon the protocol, can take place as low as the datalink level. These tasks were both placed in the transport layer, but are modular, and could be extracted and placed elsewhere. Also, the simulator was originally designed for datagram service, and since the packets do not necessarily arrive in order, it is unlikely that assembly would take place at the datalink level.

3) *The Network Layer:* The network layer is concerned with controlling the operation of the network. A key design issue is determining how packets are routed from source to destination. Another issue is how to avoid the congestion casued when too many packets are presented to the network at the same time. In the simulator, the network layer performs all of the functions related to these two aspects with the exception of that aspect of flow control which takes place at the session layer, and the recovery protocols which require some service from the datalink layer. It also activates new channels when needed and determines when packets originating at other nodes are to be discarded. The network layer is currently the most dynamic with regard to the coding of modules. Five modules currently comprise the network layer. These include relatively static modules; one module for capturing lines or channels when more capacity is required and releasing them when they are not needed; one module for the network processor and queue handling and one module for the routines which are common to most routing algorithms. This leaves two modules for the dynamic parts of the routing and flow control algorithms.

4) The Datalink Layer: The main task of the datalink layer is to take the raw transmission faciity and transform it into a line or channel that appears free of transmission errors to the network layer. It simulates the sending of the message over the channel and the delivery at the other end. When a packet is received, the datalink acknowledgment is initiated either by the piggy-back acknowledgment or by generating a datalink ackowledgment packet. As mentioned previously, the datalink level also simulates the physical layer on a statistical basis. (Entered bit eror rates are used in conjunction with a random number generator to determine if messages are corrupted.) When a line is "brought up," health packets are used to establish initial connections. Also, when a line "goes down," an active node will immediately issue health check packets to ascertain when the channel is again available.

C. Modifications

A major problem of using this system as a simulation tool for the study of packet video is that as initially designed the system did not actually transmit messages from node to node. While a "packet" carrying all the necessary describing information moved from node to node, there was no actual data in the packet. Therefore, modifications had to be made to the simulator to accommodate the video data. In the sending node, a field called "Image" which contains real image data is attached to the record "Packet_Ptr" allocated to the message generated in the session layer. There are three new modules in this layer. First, "Get_Image" puts the image data into the image field of a message generated at a specific time and node. Second, "Image_Available" checks to see if there is any image data that still need to be transmitted. If that is true, the following message, generated at that specific node, is still the image message and contains some image data. Third, "Receive_Image" collects the image data in the session layer of the receiving node when the flag "Image_Complete" is on. In module "Session_Msg_Arrive," different priorities are assigned to different messages. In module "Session_Msg_Send," some statistics are calculated including the number of lost image packets and the transmission delay for image packets.

In the original design, the transport layer simply duplicated the same packet with different assigned sequential packet numbers without actually packetizing the message. The module "Transport_Packetize" has been modified to really packetize the image data which resides in the message record queued in "Transport_Q" when it is called. The module "Transport_Reassemble" is called to reassemble these image packets according to their packet number when the flag "Image_Content" defined in "Packet _Ptr" is true. The network layer is responsible for routing and flow-control. This module was already very well developed, so the modifications to be performed here were relatively minor. In the datalink layer, in order to simulate the delivery of packets through the channel, a new packet is generated at the receiving node and the information including the image data from the transmitted packet (which will still be resident at the sending node) are copied into it. Using existing bit-error-rates, the transmission success rate can be set and bit errors can be inserted in both the data and control bits in the packet. Errors in the control bits are simulated separately as long as the error rates are consistent. If an error in the control bits occurs, the transmission is assumed to fail and retransmission will occur, again depending on the threshold of the timeout number. In addition to the modifications made to the layer modules, we had to arrange some new memory elements allocated for image messages and packets. In order to make sure the simulation is run in the steady state, the image data is made available to the network after some simulaton time has passed.

V. Interaction of the Coder and the Network

When the video data is packed and sent into a nonideal network, some problems emerge. These are discussed in the following section.

A. Packetization

The task of the packetizer is to assemble video information, coding mode information, if it exists, and synchronization information into transmission cells. In order to prevent the propagation of the error resulting from the packet loss, packets are made independent of each other and no data from the same block or same frame is separated into different packets. The segmentation process in the transport layer has no information regarding the video format. To avoid the bit stream being cut randomly, the packetization process has to be integrated with the encoder, which is in the presentation layer of the users's premise. Otherwise, some overhead has to be added into the datastream to guide the transport layer to perform the packetization in the desired manner. In order to limit the delay of packetization, it is necessary to stuff the last cell of a packet video with dummy bits if the cell is not completely full.

Every packet must contain an absolute address which indicates the location of the first block it carries. Because every block in MBCPT has the same number of bits in each pass, there is no need to indicate the relative address of the following blocks contained in the same packet. There always exists a tradeoff between packaging efficiency and error resilience. If error resilience is considerable, one packet should contain a smaller number of blocks. However, since each channel access by a station contains overhead, the packet length should be large for transmission efficiency. Fixed length packetization is used in this paper for simplicity.

Because of the structure of the coding scheme, the packets are classified into four priorities, with the packets from the first pass classified as the highest priority packets, and the packets from the fourth pass as the lowest priority packets.

This priority assignment also reflects the importance of the various packets to the reconstruction of the image sequence at the receiver. Table I shows the effect of approximately the same number of packets lost in each pass on the reconstructed error in the received sequence.

B. Error Recovery

There is no way to guarantee that packets will not get lost after being sent into the network. Packet loss can be mainly attributed to two problems. First, bit errors can occur in the address field, leading the packets astray in the network. Second, congestion can exceed the networks management ability and packets are forced to be discarded due to buffer overflow. Effects created by higher pass packet loss (like pass 4) in MBCPT coding will be masked by the basic passes and replaced with zeros. The distortion is almost invisible when viewing at video rates because the lost area is scattered spatially and over time. However, loss of low pass packets (like pass 1), though rare due to high priority, will create an erasure effect due to packetization and the effect is very objectionable.

Considering the tight time constraint, retransmission is not feasible in packet video. It may also result in more severe congestion. Thus, error recovery has to be performed by the decoder alone. In our differential MBCPT scheme, the packets from pass 4 are labeled lowest priority and form a great part

TABLE II
NUMBER OF BITS TRANSMITTED FOR EACH PASS AND THE TOTAL NUMBER OF BITS TRANSMITTED FOR EACH FRAME

Frame	Overhead	Pass1	Pass2	Pass3	Pass4	Total
1	2588	4352	8400	24248	24416	64004
2	1772	4352	5992	15232	11312	38660
3	2156	4352	7168	19432	20104	53212
4	2088	4352	6888	18760	13216	45304
5	2164	4352	7112	19600	17416	50644
6	1988	4352	6328	17920	14336	44924
7	2352	4352	7448	21896	22736	58784
8	2432	4352	7952	22512	25704	62952
9	2316	4352	7504	21336	24136	59644
10	2568	4352	7840	24528	26992	66360
11	1892	4352	6048	16856	11144	40292
12	2352	4352	7616	21728	18200	54248
13	1968	4352	6384	17584	15008	46296
14	2468	4352	7840	23128	26936	64734
15	2216	4352	9352	18088	728	34736
16	1496	4352	4536	12824	12936	36164
Total	34816	69632	114408	315672	287392	820992
Mean	2176	4352	7150	19729	17962	51312
Deviation	290	0	1094	3179	7000	10395

of the total data. These packets can be discarded whenever network congestion occurs. That will reduce the network congestion and will not cause too much degradation in quality. The erasures caused by basic pass loss are simply covered with the reconstructed values from the corresponding area in the previous frame. This remedy seems insufficient even when there is only a small amount of motion in that area. Motion detection and motion compensation could be used to find a best matched area for replacement in the previous frame.

Side information in the MBCPT decoding scheme is very important. So, this vital information is not allowed to get lost. Two methods can be used for protection. First, error control coding, like block codes or convolutional codes, can be applied in both directions along with and perpendicular to the packetization. The former is for bit error in the data field while the latter is for packet loss. The minimum distance that the error control coding should provide depends on the network's probability of packet loss, correlation of such loss and channel bit error rate. Second, from Table II, we can see that the output rate of side information and pass 1 and even pass 2 is quite steady. It seems feasible to reserve a certain amount of channel capacity to these outputs to ensure their timely arrival. That means circuit-switching can be used for important and steady data.

C. Flow Control

In order to shield the viewer from severe network congestion, there are some flow control schemes which are considered useful. It there is an interaction between the encoder and the transport layer, then the encoder can be informed about the network condition. Depending on that, the encoder can adjust its coding scheme. In the MBCPT coding scheme, if the buffer is getting full, that means that the bit generating rate is overwhelming the packetization rate and the encoder will switch to a coarse quantizer with fewer steps or loosen the threshold to decrease its output rate. In this way, smooth quality degradation is obtainable. However, this also complicates the encoder design.

It is possible to use the congestion control of the network protocols to prevent drastic quality change by assigning different priorities to packets from different passes. Ignoring the relative importance of each packet and discarding packets blindly sometimes brings disaster and can cause a session shut down. For example, if the side information gets lost it can have a severe impact on the decoding process. In the MBCPT coding scheme, side information and packets from pass 1 are assigned highest priority and higher pass packets are assigned with decreasing priority.

D. Interaction with Protocols

In the ISO model, physical, datalink and network layers comprise the lower layers which form a network node. The higher layers have transport, session, presentation and application layers and typically reside in a customer's premises. The lower layers have to do nothing about the signal processing and only work as a "packet pipe." The physical layer requires adequate capacity and low bit error rate which are determined only by technology. The datalink layer can only deal with link-management because all the mechanics, like requesting retransmission, are not feasible in packet video transmission. The network layer has to maintain orderly transmission by deleting the delay jitter with input buffering. Otherwise, it can take care of the network congestion by assigning transmission priority.

As the higher layers reside in the customer's premises, it performs all the functions of the packet video coder. The transport layer does the packetization and reassembly. The packet length can be fixed or variable. Fixed packet length simplifies segmentation and packet handling while a variable packet length can keep the packetization delay constant. The

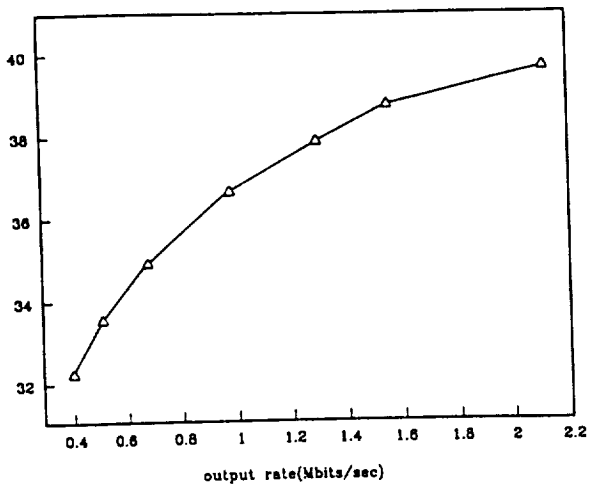


Fig. 13 PSNR versus video output rate (video transmission at 30 frames/s).

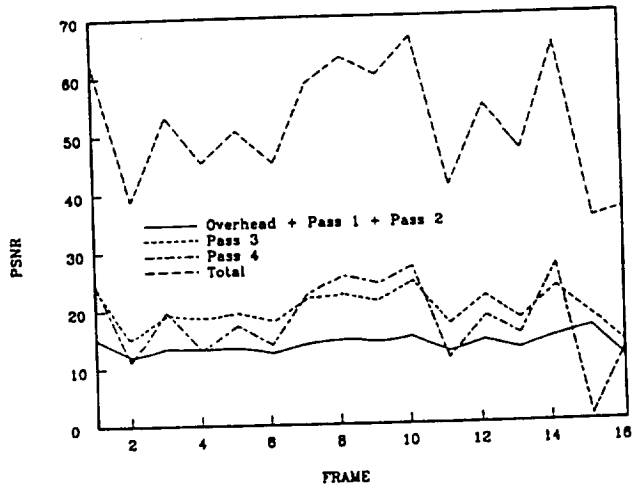


Fig. 14. Data rate of simulation sequence fames.

session layer supervises set-up and tear-down for sessions which have different types and quality. There is always a trade-off between quality and cost. The quality of a set-up session can be determined by the threshold in the coding scheme and the priority assignment for transmission. Of course, the better the quality, the higher the cost. Fig. 13 shows the tradeoff between PSNR and video output rate by adjusting thresholds. The presentation layer does most of the signal processing, including separation and compression. Because it knows the video format exactly, if any error concealment is required, it will be performed here. The application layer works as a boundary between the user and the network and deals with all the analog-digital signal conversion.

VI. Performance Results

Results obtained in this packet video simulation show that substantial compression can be obtained while maintaining high image quality through the use of this differential MBCPT scheme. The monochrome sequence used in this simulation contains 16 frames, each of size 256 × 256 pixels with 8 bits per pixel, which results in a bit rate of 15.3 Mbits/s, given a video rate of 30 frames/s. As Table II shows, the average data rates of our system is 1.539 Mbits/s. The compression rate is about 10 with a mean PSNR of 38.74 dB where PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \frac{\sum (255)^2}{\sum (x_{ij} - \hat{x_{ij}})^2}$$

Fig. 14 shows the data rate of sequence frames with side information, 4 passes and total rate. It is clear that the data rate of pass 1 is constant as long as the quantization mode remains the same. Side information and data from pass 2, even pass 3, is also relatively constant (Table III). The data rate of pass 4 is bursty and are highly uncorrelated. As pass 4 data is not essential to the reconstruction of the image, the rate profiles as shown in Fig. 14 and Table I suggest the use of a reserved channel of some sort for passes 1–3 and the side information, and perhaps a more unreliable channel for pass 4 data which comprises more than 30% of the total traffic. Such a situation can be accommodated in a variety of systems such as a token ring network or a circuit switched network with a packet-switched overlay.

TABLE III
Output bit rate for each Pass and the Total bit Rate. The Rates Were Calculated with 30 Frames/s Video Rate. The Maximum and Minimum Values are the Instantaneous Rates, Which Correspond to the Respective Maximum and Minimum Number of Bits Needed to Encode a Particular Frame in the Sequence. The Unit is Kilobits.

	Overhead	Pass 1	Pass 2	Pass 3	Pass 4	Total
Mean	65.28	130.56	214.50	591.87	538.86	1539.36
Deviation	8.70	0.00	32.82	95.37	210.00	311.85
Maximum	77.04	130.56	280.56	735.84	821.52	1990.80
Minimum	44.88	130.56	136.08	384.72	21.84	1042.08

Fig. 15 shows the PSNR for each frame in the sequence. Notice that the standard deviation of the PSNR is only 0.2 dB, which implies a substantial uniformity of quality, at least in terms of objective performance measures. If constancy with regard to some subjective criterion is desired, it would be necessary to incorporate this in the determination of the thresholds and the decision mechanism for the quad tree. In the simulation, the same threshold has been used throughout the sequence. If further flexibility, say for higher visual quality, is desired, a varying threshold can be used for different frames. That may generate a more variable bit rate.

From the difference images of this sequence, frames 1–8 seem quite motionless while frames 9–13 contain substantial motion. We adjusted the traffic condition of the network to force some of the packets to get lost and thus check the robustness of the coding scheme. Heavy traffic was set up in the motionless and motion period separately. The average packet loss percentage was 3.3%, which is considered high for most networks. Fig. 16 shows images which suffered packet losses from pass 4. As can be seen, the effect of lost packets is not at all severe, even if the lost packet rate is unrealistically high. This is because of the performance from the first three passes is relatively good and the packet from the fourth pass is not essential for reconstruction. Fig. 17 shows the case when packet loss occurs in pass 1. Clearly there are visible defects in the motion period. Further, the

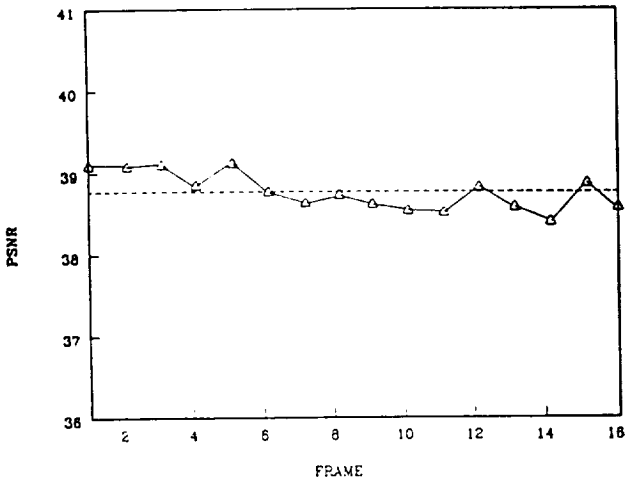


Fig. 15. PSNR of simulation sequence frames.

(a)

(b)

Fig. 16. (a) The effect of pass 4 packet loss for frame 4. (b) The effect of pass 4 packet loss for frame 10.

error will propagate to the following frames. Apparently, the replenishing scheme used here is not sufficient in areas with motion. It is believed that this inconsistency can be eliminated with a motion compensator algorithm which would find the appropriate area for replenishment and error concealment which limits the propagation of error.

VII. Conclusion

The network simulator was used only as a channel in this simulation. In fact, before the real-time processor is built, a lot of statistics can be collected from the network simulator to improve upon the coding scheme. These include transmission delays and losses from various passes under different network loads. For resynchronization, the delay jitter between received packets can also be estimated from the simulation. The environment for tomorrow's telecommunication has been described and requires a flexibility which is not possible in a circuit-switched network. With all the requirements for applying packet video in mind, MBCPT has been investigated. It is found that MBCPT has appealing properties, like high compression rate with good visual performance, robustness to packet lost, tractable integration with network mechanics and simplicity in parallel implementation. Some additional considerations have been proposed for the entire packet video system, like designing protocols, packetization, error recovery and resynchronization. For fast moving scenes, the differential MBCPT scheme seems insufficient. Motion compensation, error concealment of even attaching function commands into the coding scheme are believed to be useful tools to improve the performance and will be the direction of future research.

(a)

(b)

Fig. 17. (a) The effect of pass 1 packet loss for frame 3. (b) The effect of pass 1 packet loss for frame 9.

References

[1] W. Verbiest and L. Pinnoo, "A variable bit rate codec for asynchronous transfer mode networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 761–770, June 1989.

[2] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 771–781, June 1989.

[3] J. Darragh and R. Baker, "Fixed distortion subband coding of images for packet-switched networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 789–800, June 1989.

[4] F. Kishino, K. Manaabe, Y. Hayashi, and H. Yasuda, "Variable bit-rate coding of video signals for ATM networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 801–806, June 1989.

[5] G. Karlsson and M. Vetterli, "Packet video and its integration into network architecture," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 739–751,

June 1989.
[6] M. C. Rost, "Data compression using adaptive transform coding," Ph.D. dissertation, Univ. Nebraska-Lincoln, 1988.
[7] M. C. Rost and K. Sayood, "A progressive data compression scheme based on adaptive transform coding," in *Proc. 31st Midwest Symp. Circuits Syst.* St. Louis, MO, Aug. 1988, pp. 912–915.
[8] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. Robbins, "Performance analysis of statistical multiplexing for packet video sources," in *Proc. GLOBECOM–87*, Tokyo, Japan, Nov 1987, pp. 47.8.1–47.8.10.
[9] K. Sloan Jr., and S. Tanimoto, "Progressive refinement of Raster scan images," *IEEE Trans. Comput.*, vol. COM-28, pp. 871–874, Nov. 1979.
[10] H. M. Dreizen, "Content-driven progressive transmission of Grey-scale images," *IEEE Trans. Commun.*, vol. COM-35, pp. 289–296, Mar. 1987.
[11] D. Vaisey and A. Gersho, "Variable block-size coding," in *Proc. ICASSP*. pp. 1051–1054, Apr. 1987.
[12] Y. S. Ho and A Gersho, "Variable-rate multi-stage vector quantization for image coding," in *Proc. ICASSP*, vol. 2, 1988, pp. 1156–1159.
[13] D. Nelson, K. Sayood, G. Ankenman, and H. Chang, "Pnetsim system programmer's manual," Univ. Nebraska-Lincoln, Final Rep. ARMY Contract DAAB07-85-K-K535, Dec. 1986.

Yun-Chung Chen was born in Kaoshkung, Taiwan, in 1961. He received the B.S. degree from the National Taiwan University, Taiwan, in 1983 and the M.S. degree in electrical engineering from University of Nebraska-Lincoln in 1989. He is currently pursuing the Ph.D. degree at the University of Nebraska-Lincoln.

Khalid Sayood (S'78–M'79–S'80–M'82) was born in Pakistan in 1956. He received his undergraduate education at the Middle East Technical University, Ankara, Turkey, and the University of Rochester, Rochester, NY. He received the B.S. and M.S. degrees from the University of Rochester, and the Ph.D. degree from Texas A&M University College Station, TX, in 1977, 1979, and 1982 respectively, all in electrical engineering.

He joined the Department of Electrical Engineering at the University of Nebraska-Lincoln, in 1982, where he is currently serving as Professor. His current research interests include data compression, joint source/channel coding, communication networks, and biodmedical applications.

Dr. Sayood is a member of Eta Kappa Uu and Sigma Xi.

Don J. Nelson (S'52–A'53–M'58–SM'84) was born in Pilger, NE, on August 17, 1930. He received the B.Sc. degree with distinction and the M.Sc. degree from the University of Nebraska in 1953 and 1958, respectively, and the Ph.D. degree from Stanford in 1962, all in electrical enginering.

From 1953 until 1955, he was a member of Technical Staff at Bell Laboratories, and served as a 2nd and 1st Lt. in the Air Force. He has been on the staff at the University of Nebraska since 1955, directing the research computing facility from 1963 to 1972. He is currently a Professor of Electrical Engineering and Computing Science, and Co-Director of the Center of Communication and Information Science. His research interests include communication network simulation, simulation of continuous systems, estimation, adaptive algorithms and error analysis of integration algorithms.

Dr. Nelson is a member of ACM, Eta Kappa Nu, Pi Mu Epsilon, Sigma Tau, Sigma Xi, and Tau Beta Pi.

Recursively Indexed Differential Pulse Code Modulation*

Khalid Sayood† and Sangsin Na‡

January 21, 1993

Abstract

We study the performance of a DPCM system with a recursively indexed quantizer (RIQ) under various conditions, with first order Gauss-Markov and Laplace-Markov sources as inputs. We show that when the predictor is matched to the input, the proposed system performs at or close to the optimum entropy constrained DPCM system. We also show that the if we are willing to accept a 5% increase in the rate, the system is very forgiving of predictor mismatch.

1 Introduction

Differential pulse code modulation (DPCM) is often used to efficiently convert an analog source such as speech, music or images into a digital form for communication or storage. Its efficiency is due to the exploitation of the memory in a source by the use of a predictor, which estimates the present source sample to be encoded, based on the quantized previous source samples. The performance of DPCM depends on two factors

1. How well the predictor exploits the source memory, i.e. , how closely it can estimate the actual source samples.
2. How well the quantizer is matched to the prediction error (the quantizer input).

In order to maximize the goodness of prediction, the predictor is usually chosen based on the statistical properties of a given source. However, many physical sources, such as those listed above, exhibit statistically varying local properties which are usually quite distinct from their global ones. If this is the case and DPCM happens to operate on a segment whose statistics differ from the global ones, it operates in a mismatched state, which results in additional degradation in the reproduction [1].

Various schemes have been devised to handle this mismatch between the source and the predictor. These involve some form of adaptation of the quantizer and/or the predictor, by which DPCM quickly follows up the changing statistics of the source and prevents overloading of the system. However, this quick response of DPCM is not without its cost: the adaptation requires more implementation and operational complexity.

Matching the quantizer to the statistics of the prediction error is even more difficult, as the quantizer structure itself effects the statistical properties of the prediction error. One way to obtain the statistics of the prediction error process is through the use of an orthonormal expansion [2, 3, 4, 5]. This has been used to optimize the quantizer through an iterative procedure [4, 6, 7, 8]. However, under operational circumstances this might not be a viable option.

*This work was supported by NASA Lewis esearch Center under grant NAG 3-806, and The Goddard Space Flight Center under grant NAG 5-1615

†Department of Electrical Engineering, University of Nebraska - Lincoln, Lincoln, NE 68588-0511

‡Department of Electronic Engineering, Ajou University, Suwon, Korea

In this paper we study the performance of a DPCM system operating on a Gauss-Markov source and a Laplace-Markov source. The DPCM system considered here, called a recursively indexed DPCM, consists of a uniform quantizer with infinitely many output levels, a recursively indexed binary encoder, and a first-order linear predictor. The quantizer is designed simply by specifying its step-size, and the predictor is non-adaptive. The Gauss-Markov and Laplace-Markov sources are chosen because of their use in the modeling of physical sources. The goal is to observe the rate distortion performance of this system, as well as the performance degradation when the predictor and the source are mismatched. We compare the rate distortion performance of the proposed system to the optimum results in the literature [6], where the quantizer was optimized using the iterative procedure mentioned above.

The simulation results show that the rate distortion performance of the proposed system achieves or comes very close to the optimum performance at all rates studied. In the case of mismatch the simulation results show that only a 5 percent increase in the rate allows a rather wide range of the predictor mismatch to first-order Gauss-Markov and Laplace-Markov sources. They agree with a result in [9] reported for a 2-level optimized DPCM that the predictor coefficient does not significantly effect the optimality of DPCM. But unlike in [9] it is observed that a lower predictor coefficient than the source correlation coefficient is better for the low rate case.

In Section 2 a recursively indexed binary encoder is discussed. In Section 3 DPCM is briefly reviewed and the DPCM mismatch problem is posed. In Section 4 the performance of a recursively indexed DPCM is considered and numerical results are presented. Conclusions follow in section 5.

2 A Recursively Indexed Binary Encoder

The DPCM system discussed here uses a quantizer with infinitely many output levels. This requires binary encoding of a countably infinite alphabet, which poses obvious problems in design and operation. An obvious and reasonable approach is first to represent the input alphabet using only finite many symbols and then to encode these symbols either using a fixed-to-fixed or fixed-to-variable length encoding. A recursively indexed binary encoder is used for just this purpose.

The recursively indexed binary encoder considered in this paper is a two stage binary encoder: recursive indexing followed by an optimum (the minimum average codeword length) symbol-to-variable length binary encoder for the output of recursive indexing.

Recursive indexing is a mapping of a countable set to a collection of sequences of symbols from another set of finite size [10, 11]. Given a countable set $A = \{a_0, a_1, \ldots\}$ and a finite set $B = \{b_0, b_1, \ldots, b_{M-1}\}$ of size M, the recursive indexing of A by B is a mapping I of A to the collection of all sequences of symbols from B such that

$$I(a_i) = \underbrace{b_{M-1}b_{M-1}\ldots b_{M-1}}_{q \text{ times}}b_r \text{ if } i = q(M-1)+r \tag{1}$$

where q and r are the quotient and remainder of i when divided by $M-1$. Set B is called the representation set. Defined as such, recursive indexing is a one-to-one mapping, a symbol-to-variable length, M-ary, prefix-free code and therefore uniquely and instantaneously decodable.

Since the second stage of the recursively indexed binary encoder is an optimum symbol-to-variable length encoder (the Huffman algorithm is used to design such), the statistic of the representation symbols must be computed. For this purpose we first compute the number of representation symbols needed to describe a typical source sequence $X_1, X_2, \ldots X_n$ of the length n from set A. Define $p_k = Pr(X = a_k)$, then the number n_0 of the occurrences of symbol b_0 is computed as follows. Observe that b_0 occurs once whenever $a_0, a_{M-1}, a_{2M-2}, \ldots, a_{k(M-1)}, \ldots$ occur. The number of times these symbols occur is given by $np_0, np_{M-1}, np_{2M-2}, \ldots, np_{k(M-1)}, \ldots$ and so on. Therefore,

$$n_0 = n \sum_{k=0}^{\infty} p_{k(M-1)}. \tag{2}$$

In a similar manner the number n_j of the occurrences of symbol b_j are found to be:

$$\begin{aligned} n_j &= n \sum_{k=0}^{\infty} p_{k(M-1)+j} \text{ for } j = 0, 1, \ldots, M-2. \\ n_{M-1} &= n \sum_{k=0}^{\infty} \sum_{j=0}^{M-1} k p_{k(M-1)+j}. \end{aligned} \tag{3}$$

From these it is seen that on the average the number of representation symbols needed for n source symbols is

$$\sum_{j=0}^{M-1} n_j \tag{4}$$

Therefore, the average number of representation symbols to represent one source symbol is

$$\frac{1}{n} \sum_{j=0}^{M-1} n_j = 1 + \frac{n_{M-1}}{n} = 1 + \sum_{k=0}^{\infty} \sum_{j=0}^{M-1} k P_{k(M-1)+j} \tag{5}$$

It is convenient to define the above expression as the expansion factor, denoted e, of the recursive indexing I. It is the factor by which one source symbol is expanded by the recursive indexing. The relative frequency q_j of representation symbol b_j, is computed as follows:

$$q_j = \frac{n_j}{n}. \tag{6}$$

An optimum symbol-to-variable length binary encoder after the recursive indexing takes one representation symbol at a time and produces the corresponding binary sequence from a set of variable length code words. It is designed for example using the Huffman algorithm. Then its rate R_{RI} , the number of binary digits per representation symbol, is bounded by

$$H(B) \leq R_{RI} < H(B) + 1 \tag{7}$$

symbol-to-variable length code by the Huffman algorithm it is observed that rate R is almost equal to $H(B)$, the lower bound, when M is large.

The overall rate of the recursively indexed binary encoder then is bounded

$$eH(B) \leq R < eH(B) + e \tag{8}$$

Note that e approximately equals 1 if M is large.

3 DPCM with Recursively Indexed Binary Encoding

3.1 Source and DPCM

Let us consider the encoding by a DPCM system of a first-order Gauss-Markov process,

$$X_k = \rho X_{k-1} + W_k \tag{9}$$

where W_k is an independent identically distributed Gaussian with mean zero and variance σ_w^2 . The source correlation coefficient ρ is between -1 and 1.

The DPCM system consists of a quantizer, a predictor and a binary encoder In a typical operational cycle the difference Z_k between the source output X_k and its prediction $\hat{X}_k$ is quantized by quantizer Q yielding $Q(Z_k)$, which in turn is binary encoded. The predictor considered in this paper is a first-order linear predictor and therefore it is given by bY_{k-1} for some constant b.

We will say that DPCM is matched to the source if the source correlation ρ equals the predictor coefficient b and that it is mismatched otherwise.

The binary encoder in ordinary DPCM is either a fixed-to-fixed length or a fixed-to-variable length binary encoder. In the former the binary encoder takes the index of the quantizer output level, produces its binary representation and sends the binary sequence through the channel. In the latter blocks of quantizer outputs are buffered and (usually) entropy-encoded.

The performance of a DPCM system will be measured by distortion and rate. The distortion incurred is defined to be

$$D = \limsup_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} E\Big\{(X_k - Y_k)^2\Big\}. \tag{10}$$

It is well-known that the error of the DPCM system is that incurred by the quantizer alone and nowhere else, i.e., $X_k - Y_k = Z_k - Q(Z_k)$ Hence the distortion can be rewritten as

$$D = \limsup_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} E\Big\{(Z_k - Q(Z_k))^2\Big\}. \tag{11}$$

The rate is defined to be the average number of binary digits used to transmit one source symbol. In case of a fixed-to-fixed length binary encoder is given by $\lceil \log_2 N \rceil$, where N is the number or quantizer output levels. In case of a fixed-to-variable length entropy encoder it is approximately H(Q), the entropy of the quantizer output process.

3.2 Recursively Indexed DPCM

The DPCM system considered in this paper is a recursively indexed DPCM system. It is different from an ordinary system in the following two ways:

1. The quantizer Q is an infinite level uniform quantizer with the thresholds being the mid-points of output levels.

2. The binary encoder is a recursively indexed fixed-to-variable length encoder.

The quantizer with infinitely-many output levels uniformly spaced yields granular distortion only. The magnitude of the distortion is bounded by $\Delta/2$. Therefore, no matter how large the input to the quantizer is, due to bad prediction, its output is at most $\Delta/2$ different from its input. Since an unoverloaded quantized value is available to the predictor at the next prediction, the system can track the source output, thereby yielding lower prediction errors. Due to this quick response the system does not have catastrophic error propagation, which DPCM with a finite number of quantizer output levels has when a pathological source sequence is encountered.

As discussed in Section 2, a recursively indexed binary encoder is necessary because the quantizer used has infinitely many output levels. We note that the encoder is not necessarily an entropy-encoder for the quantized process.

The distortion for this system is given by (11), while the rate is simply the entropy of the representation symbols multiplied by the expansion factor.

4 Simulation Results

To test this system first order Gauss-Markov and Laplace-Markov random number generators with correlation coefficient ρ were used. The Laplace-Markov process was defined as in [6]. The variance of the innovation sequence was chosen so as to get a source variance of unity. For each realization of the process 100,000 samples were used.

4.1 Rate-Distortion Performance

To obtain the rate-distortion performance, the predictor was matched to the source correlation coefficient and the step-size was varied. Each step-size Δ generated a distortion-rate pair which was then plotted. The results were overlaid on the optimum results from [6]. The results for $\rho = 0.8$ are plotted in Figures 1 and 2.

For the Gauss-Markov source the rate distortion performance of the proposed system achieves (or almost achieves) the optimum performance for the entire range of rates. This is true for both the experimentally obtained rates from [6] as well as the asymptotic results. Recall that for the optimum results the quantizer was designed using a relatively complex iterative procedure, while for the proposed system the quantizer was designed by simply specifying the step-size.

For the Laplace-Markov source, the proposed scheme again performs as well as the optimum scheme for almost all rates as far as the experimentally obtained results are concerned. However, the proposed scheme provides *better* results than the asymptotic results in [6]. We presume this is due to an error in the asymptotic results (or our interpretation of them!).

Similar results were obtained for ρ= 0.5, and 0.2 for both sources. The results seem to indicate that the problem of matching the quantizer with the input statistics can be easily resolved by the use of a recursively indexed quantizer. The quantizer can be easily designed by simply specifying the step-size Δ, which in turn can be specified based on the distortion requirements.

4.2 Performance Under Mismatch Conditions

To investigate the effect of mismatch between the predictor and model coefficients we used three values for the spacing Δ of the uniform quantizer: $2.5\sigma_x, 1.5\sigma_x, 0.2\sigma_x$. These correspond to low, medium, and high resolution (rate) quantizations, respectively. For each spacing for uniform- quantization, the source sequence is applied to the recursively indexed DPCM system with various values for the predictor coefficient. The quantizer output sequence is fed to the recursively indexed binary encoder with M representation symbols. The value of M ranges from 5 to 31. The results are shown in Figure 3, where the horizontal and vertical axis are respectively the predictor coefficient and the product of the rate and distortion.

For the Gauss-Markov source, for low rate ($\Delta = 2.5\sigma_x$) the distortion is about 0.44. The best performance is obtained when the predictor coefficient is around 0.6, where the rate is 0.58 bits/sample. This value is far below the source correlation 0.8. Note that in [4], the best performance was reported around 0.815, slightly higher than the source correlation. If 5% increase in rate is allowed, then Figure 3 shows that the predictor coefficient can be anywhere between 0.4 and 0.8.

For medium rate ($\Delta = 1.5\sigma_x$) the distortion is about 0.186, while $\Delta^2/12$ is 0.1875. The best predictor coefficient is around 0.7. The rate for this value of Δ and b is 1.12 bits/sample. The value of b is again lower than the source correlation. Again a 5% increase in rate allows the predictor coefficient anywhere between 0.48 and 0.88. For high rate ($\Delta = 0.2\sigma_x$) the distortion is about 0.00333, while $\Delta^2/12$ is 0.00333. The best predictor coefficient is around 0.8, where the rate is 4.2 bits/sample, and can range from 0.0 to 0.99 if 5% increase in rate is allowed.

We observe that the best predictor coefficient moves from below closer to the source correlation as resolution increases. This may be because for larger values of Δ the quantization noise tends to be magnified when the predictor coefficient is larger[1]. Also we note that the distortion expression ($\Delta^2/12$) is quite accurate even for large Δ. Similar results are observed for the Laplace-Markov source.

4.3 Effect of the Alphabet Size M

Finally we look at the size of the representation alphabet on the rate. Note that the larger the representation alphabet is for a certain value of Δ the less likely it is that the encoder will enter the recursive mode. This implies that given a value of Δ larger values of M will tend to lower the value of the expansion factor e closer to 1, thus lowering the rate. If this is a very strong effect, recursive indexing looses some of its charm, as the smaller alphabet size of the reproduction alphabet makes it more amenable to entropy coding. The recursively indexed DPCM system was simulated with representation alphabet sizes 7,9,11,13, and 15. The results for a Laplace-Markov source are shown in Figure 4. Note that at low rates (large values of Δ)there is no difference between these sizes. At higher rates, there is noticeable difference as we increase the alphabet size from 7 to 11. After that point there is very little improvement obtained in the rate when the alphabet size is increased.

5 Conclusions

Recursively indexed DPCM, which features an infinite level quantizer coded with a finite alphabet entropy coder has been shown to be an efficient encoder for first order Gauss-Markov and Laplace-Markov sources. The use of a recursively indexed quantizer in a standard DPCM system seems to provide a solution both to the problem of matching the quantizer to the prediction error statistics, and the problem of exactly matching the predictor to the source, at least for these simple sources.

References

[1] N.S. Jayant and P. Noll. *Digital Coding of Waveforms.* Prentice-Hall, 1984.

[2] L. D. Davisson. Information Rates for Data Compression. In *IEEE WESCON.* IEEE, 1968.

[3] D. Slepian. On Delta Modulation. *Bell Syst. Tech. J.*, 45:2101–2137, December 1972.

[4] D. S. Arnstein. Quantization Error in Predictive Coders. *IEEE Transactions on Communications*, COM-23(4):423–429, April 1975.

[5] M. Naraghi-Pour and D. L. Neuhoff. Convergence of the Projection Method for an Autoregressive Process and a Matched DPCM Code. *IEEE Transaction on Information Theory*, 36:1255–1264, Nov 1990.

[6] N. Farvardin and J. W. Modestino. Rate-Distortion Performance of DPCM Schemes for Autoregressive Sources. *IEEE Transactions on Information Theory*, IT-31:402–418, May 1985.

[1]Thanks to Dr. J. Johnston for his suggestion

[7] N. L. Gerr and S. Cambanis. Analysis of Adaptive Differential PCM of a Stationary Gauss-Markov Input. *IEEE Transactions on Information Theory*, IT-33:350–359, May 1987.

[8] E. Janardhanan. Differential PCM Systems. *IEEE Transactions on Communications*, COM-27:82–93, January 1989.

[9] M. Naraghi-Pour and D. L. Neuhoff. Mismatched DPCM Encoding of Autoregressive Processes. *IEEE Transactions on Information Theory*, pages 296–304, March 1990.

[10] K. Sayood and S. Na. Recursively Indexed Quantization of Memoryless Sources. *IEEE Transactions on Information Theory*, IT-38, November 1992.

[11] K. Sayood and S. Na. Recursive Quantization. In *24th Annual Asilomar Conference on Circuits, Systems, and Computers*, November 1990.

An Image Compression Technique for Use on Token Ring Networks

B. Gorjala and K. Sayood *
Department of Electrical Engineering
and
Center for Communication and Information Science
University of Nebraska - Lincoln
Lincoln, NE 68588-0511
Tel: (402) 472-6688
Fax: (402) 472-4732
email: ksayood@eecomm.unl.edu

G. Meempat
~~NEC~~ BELLCORE
Trenton NJ

August 27, 1992

Abstract

We present a low complexity technique for compression of images for transmission over local area networks. The technique uses the synchronous traffic as a side channel for improving the performance of an ADPCM based coder.

*Supported by the NASA Goddard Space Flight Center(NAG 5-916 and NAG 5-1615).

1 Introduction

The use of local area networks makes it possible to more easily implement algorithms that require the use of a "side channel". In this paper we present an ADPCM (Adaptive Differential Pulse Code Modulation) based codec which can be conveniently implemented on LANs.

Adaptive Differential Pulse Code Modulation (ADPCM) is a very popular compression technique because it is easy to implement, has low processing overhead, and relatively good fidelity. This has made it the algorithm of choice in speech compression applications, and as a second stage for subband coding and transform coding techniques. However, ADPCM image compression is far from ideal. The most obvious drawback is poor edge reconstruction. ADPCM cannot track sudden changes in image statistics, and this can cause substantial edge distortion in the reconstructed image. A modified ADPCM scheme was presented in [1] which relied on the use of side information to prevent edge degradation. The technique is well suited for implementation on token ring networks.

In this paper we describe the implementation of this scheme in a token ring network environment. The paper is organized as follows. The next section gives a brief overview of the aspects of token ring networks that are of interest here. The modified ADPCM scheme is briefly described in the following section. Then we describe the implementation of the proposed algorithm on a token ring network and present simulation results.

2 Token Ring Networks

In a token ring network, nodes are arranged logically in a ring with each node transmitting to the next node around the ring. Each node simply relays the received bit stream from the previous node to the next node with at least one bit delay. The token is defined as a special bit pattern which circulates on the ring whenever all the stations are idle. Whatever node has the token is allowed to transmit a packet. When the packet has been transmitted the token is passed on to the next node. That is, whenever the node that is currently transmitting a packet finishes the transmission, it places the token, for example 01111110, at the end of the packet. When the next node reads this token, it simply passes the token if it has no packets to send. If it does have a packet to send it inverts the last token bit, in our example turning the token to 01111111. The station or node then breaks the interface connection and enters its own data onto the ring.

The token ring supports two classes of traffic;

1. Synchronous Traffic: A class of data transmission service whereby each requester is pre-allocated a maximum bandwidth and guaranteed a response time not to exceed a specific delay.

2. Asynchronous Traffic: A class of data transmission service whereby all requests for service contend for a pool of dynamically allocated ring bandwidth and response time.

A set of timers and several parameters are used to limit the length of time a station may transmit messages before passing the token to the next station, and the duration of information transmission of each class within a station [2]. Each station maintains two timers, the Token_Rotation_Timer (TRT) and the Token_Holding_Timer (THT). The TRT at node j is used to time the interval taken by the token to circulate around the ring starting at node j. When node j recaptures the token, the value of TRT is assigned to THT and TRT is reset. When the network is initialized, the stations decide on the value of a target token rotation time (TTRT), so that the requirements for maximum access time are met. The upper bounds on the maximum and average token rotation time have been studied in [3]; the results show that the token rotation time cannot exceed twice the value of TTRT, while the average rotation time is not greater than TTRT. The extension to several priority classes is obtained by introducing a target rotation time for each class, and by using that value to check whether or not the station is allowed to transmit frames of that class.

If a station captures the token before its TRT reaches the value of TTRT, it is called an *early* token. If it captures the token after the TRT has exceeded the value of TTRT, it is called a *late* token. An *early* token may be used to transmit both synchronous and asynchronous traffic , while a *late* token may only be used for synchronous traffic. The difference between TTRT and TRT will be the available bandwidth for the asynchronous information. The

amount of time a station can transmit is limited by THT.

In the following section we describe an image compression scheme which takes advantage of lighter loads on the network to provide side information to the receiver as asynchronous traffic. This side information is then used to increase the quality of the reconstructed image.

3 Edge Correcting DPCM

The proposed ADPCM system uses a two-bit Robust Jayant quantizer [4, 5]. This is a uniform quantizer whose step-size $\Delta(k)$ is adapted based on the previous quantizer output level $H(k-1)$ according to the following recursion [6]

$$\Delta(k) = [M(H(k-1))\Delta(k-1)]^{\beta}$$

where $\beta = 1 - \epsilon^2$, $\epsilon \leftarrow 0$, and $M(1) = 0.8, M(2) = 1.6$, $H(k) = 1$ if the output falls into the inner levels of the quantizer and $H(k) = 2$ if the output is one of the outer levels of the quantizer. As the information about which level of the quantizer was used in the previous sample, is available to both the transmitter and the receiver, the adaptation does not require the transmission of any side information.

The Jayant quantizer is designed to track the variance of the quantizer noise by changing the step size $\Delta(k)$. Since edges are regions where the statistics change rapidly, it follows that the step size will expand repeatedly

when an edge is encountered. This fact is made use of in the following rule to detect edges:

> An edge is detected when the step size of the Jayant quantizer expands more than P times in succession, P ¿ 1.

The value of P should be small to reduce detection delay; a value of two seems to work well. As both transmitter and receiver have the same information both transmitter and receiver will detect edges at the same time. Once an edge has been detected the proposed scheme uses an embedded quantizer to quantize the quantization error and transmit this value over a side channel. The use of an embedded quantizer was first proposed by Goodman and Sundberg [7] for use over a noisy channel. In [1] the issue of how a side channel could be configured was left open. We address this issue in the context of token ring networks in the following section.

4 ADPCM and the Token Ring Network

As mentioned earlier, the traffic in the token ring network is divided into synchronous and asynchronous traffic. We use the regular ADPCM output as the synchronous traffic and the output of the embedded quantizer as the asynchronous traffic. Thus the side channel simply consists of the asynchronous traffic. The reasoning behind this approach is that the system cannot afford to lose the regular ADPCM output which also has timing constraints. The

side information is not as critical, because the image can be reconstructed at the receiver without the side information, albeit with some degradation.

In the analysis of a token protocol, it is generally assumed that the queues of asynchronous messages to be sent are heavily loaded, so that messages are always available for transmission. In our case the asynchronous information queue will not be heavily loaded because the side information needs to be sent only when there is an edge.

The size of the packet for synchronous traffic is fixed. Whenever the node captures an early token, the size of the packet will be increased to match the available capacity and the regular information followed by the side information, if present, will be sent. The most recent side information will be transmitted in the bandwidth available for asynchronous traffic. If there is any side information left after transmission, it will be discarded.

Whenever the receiver receives an increased size packet it takes the bits received after the regular size of the packet as side information. This side information is added to the corresponding most recent "edge" pixels.

5 Simulation of Proposed Scheme

A fifty node token ring network was simulated to test the proposed system. The parameters used in the simulation are given in Table 1.

The system is assumed to work under the following general conditions

- The packet arrival process at each node follows a Poisson distribution.

Number of nodes	50
Bit traveling speed	200,000 met/msec
Distance between nodes	100 meters
Data generation rate	11,000 bits/sec
Packet size for synchronous information	1540 bits
Time taken by node to read the data	$10\mu sec$
Channel capacity of coaxial cable	12,000 bits/msec

Table 1: Simulator parameters

The actual image information is taken at node 1 with regular ADPCM output arriving into one buffer, and the side information into a separate buffer.

- The message transmitted transmitted by each station belong to two classes, i.e. asynchronous and synchronous messages.
- The access mechanism is based on the timed token approach, but different classes of asynchronous messages are not considered.
- The queues of asynchronous messages are not heavily loaded.
- When the network is initialized, the token rotation will only allow the transmission of synchronous messages; the second token rotation will allow both synchronous and asynchronous messages.

Two types of simulations were performed.

1. Messages transmitted at all nodes consisted of both synchronous and asynchronous messages.

2. Only synchronous messages were transmitted at all nodes.

Load versus delay and throughput versus delay characteristics were plotted for both cases and are shown in figures 1 and 2. Load is defined as the inverse of the mean inter-arrival time λ. The graph in Figure 1 shows that at a particular value of the load, the average delay of a packet in the network with both classes of traffic is more compared to when only synchronous messages are transmitted. This is especially true at low loads; as the traffic increases there is not much difference in the delay for the two cases. This is because the network will not have enough bandwidth available for asynchronous traffic when the traffic is busy.

The token ring network transmitting both synchronous and asynchronous messages provides better delay versus throughput characteristics. Here again at large values of throughput, there is not much difference between the curves. The reason for the better throughput versus delay characteristics is that at low loads, the network can utilize the channel more efficiently by transmitting asynchronous messages whenever the bandwidth becomes available.

The two images shown in Figure 3 were used to test the proposed approach and the results obtained at different network loads are shown in Table 3 and Table 2.

The first two entries in these tables were obtained by operating the net-

Run	Load	Delay msec	Throughput	Rate bpp	PSNR dB
1	.226	177.3	0.8033735	2.011	33.33
2	.185	147.8	0.8033297	2.014	33.36
3	.156	123.3	0.8032793	2.057	34.76
4	.136	105.3	0.8032306	2.126	35.69
5	.119	89.19	0.8030134	2.221	36.05
6	.107	73.17	0.8019965	2.223	37.22
7	.097	58.35	0.8000950	2.237	37.59
8	.085	42.82	0.7967703	2.238	37.62
9	.081	38.87	0.7956204	2.238	37.62

Table 2: Results obtained at different network loads for couple image

Run	Load	Delay msec	Throughput	Rate bpp	PSNR dB
1	.254	194.1	0.8033794	2.002	29.13
2	.169	135.3	0.8033345	2.016	29.22
3	.156	123.6	0.8033151	2.039	29.32
4	.145	114.4	0.8033025	2.075	29.78
5	.127	97.5	0.8032005	2.183	30.68
6	.107	70.3	0.8019068	2.227	30.90
7	.092	51.8	0.7989234	2.237	31.02
8	.081	38.4	0.7954938	2.237	31.02

Table 3: Results obtained at different network loads for aerial image

work at high loads which is in the unstable region. At these high loads almost every node will have a packet to send, and there was no bandwidth available for side information. As the load was decreased, more and more side information was transmitted, providing a better reconstructed image at the receiver. In this simulation, at a load of around 0.09, there is enough bandwidth available for node 1 to transmit all the side information. Further reduction of the load did not have any effect on the quality of the reconstructed image.

Error images for the couple image were obtained at four different network loads and are shown in Figure 4. The error image without any side information is shown in Figure 4a for comparison. For the image shown in Figure 3b, side information was sent in the areas of the woman's hands, the woman's left knee and in some portions of the couples heads. In figure 4c the edge errors are corrected in the region of the womans hands, the man's shoulder, the photo frame, and the couple's heads. Some of the edge errors at the man's legs are also corrected. But in this case edge errors are present at the woman's left knee. In the image shown in Figure 4d all edge errors are corrected except a few errors at the intersection of the man's leg and chair. For the image shown in Figure 4e all side information was transmitted.

6 Conclusion

We have presented a low complexity scheme which can be used for transmitting images over local area networks. Because of its low complexity the

scheme can be operated at high rates and may be suitable for applications which require low delay.

References

[1] S.M. Schekall and K. Sayood. An Edge Preserving DPCM Scheme for Image Coding. In *Proceedings Midwest Symposium on Circuits and Systems*, pages 904–907. IEEE, 1988.

[2] A. Valenzano, P. Montuschi, and L. Ciminiera. On the Behavior of Control Token Protocols with Asynchronous and Synchronous Traffic. In *Proceedings IEEE INFOCOM*. IEEE, September 1989.

[3] K.C. Sevcik and M.J. Johnson. Cycle Time Properties of the FDDI Token Ring Protocol. *IEEE Transactions on Software Engineering*, SE-13:376–385, March 1987.

[4] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[5] N.S. Jayant. Adaptive Qunatization with One Word Memory. *Bell System Technical Journal*, pages 1119–1144, September 1973.

[6] D.J. Goodman and R.M. Wilkinson. A Robust Adaptive Quantizer. *IEEE Transactions on Communications*, pages 1362–1365, November 1975.

[7] D.J. Goodman and C.E. Sundberg. Combined Source and Channel Coding for Variable Bit-Rate Speech Transmission. *Bell System Technical Journal*, 62:2017–2036, September 1983.

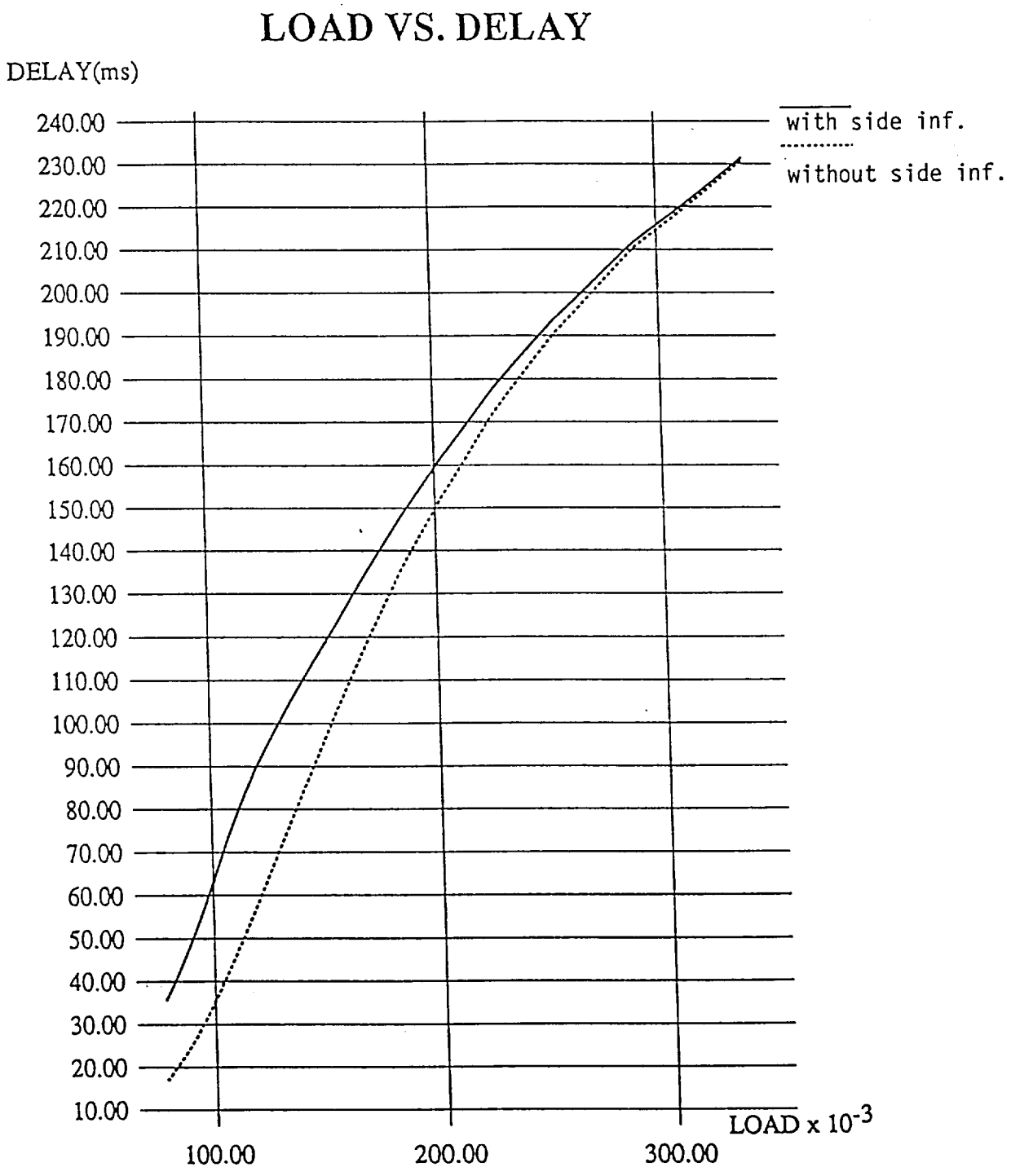


Figure 1 Load versus Delay for the Token Ring

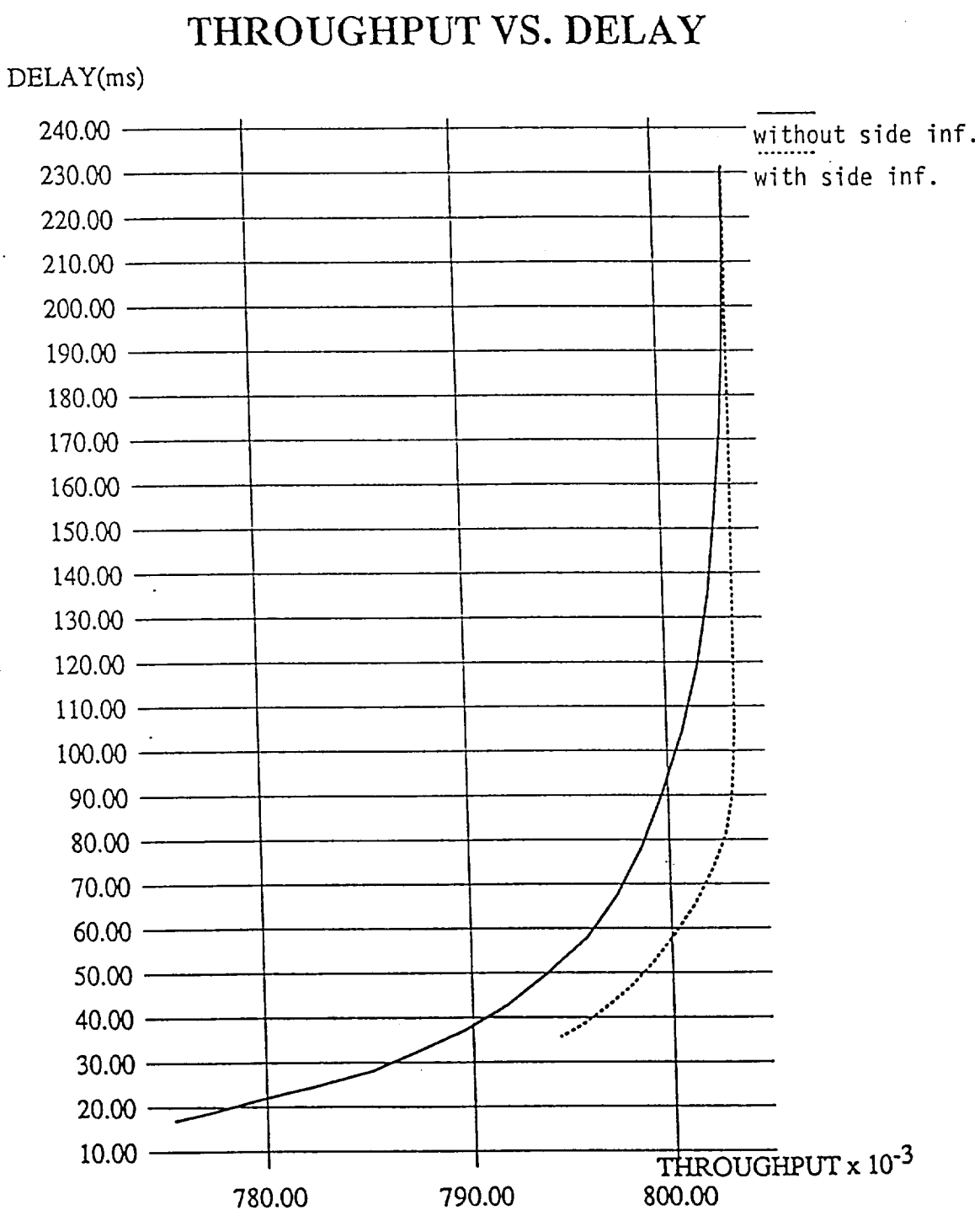


Figure 2 Throughput versus Delay for the Token Ring

C-2

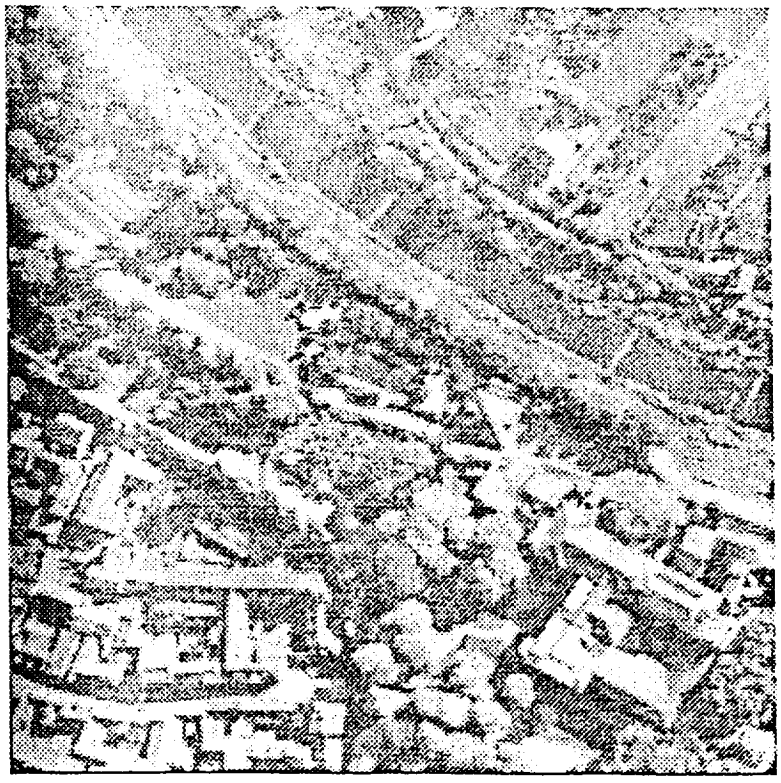
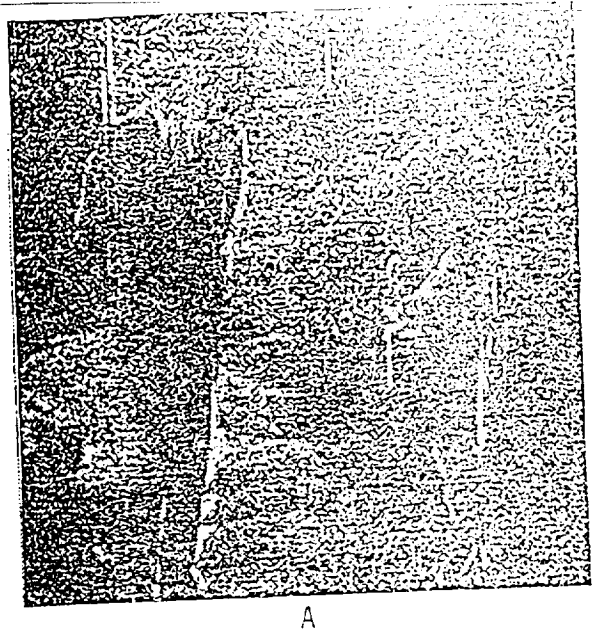
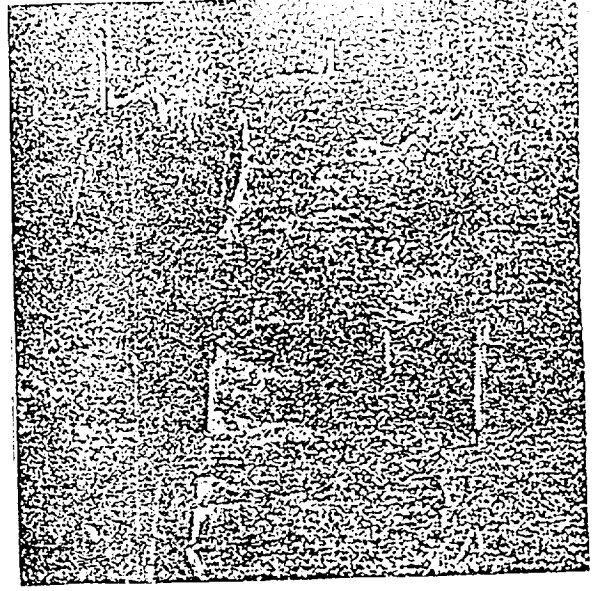


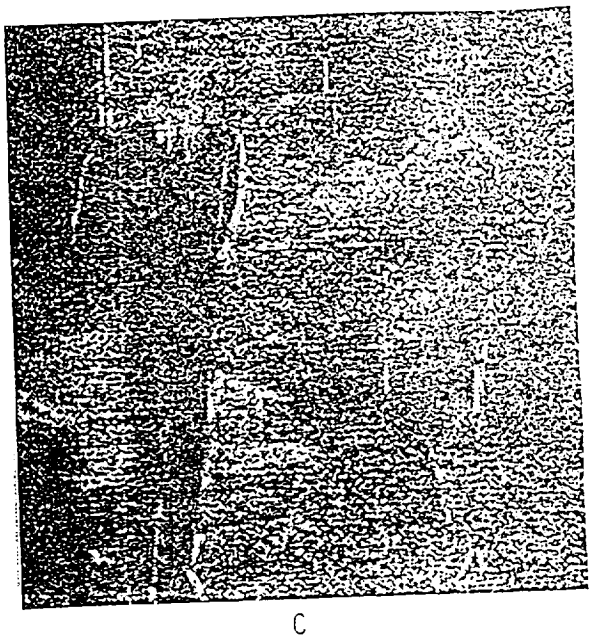
Figure 3. Images used in simulations



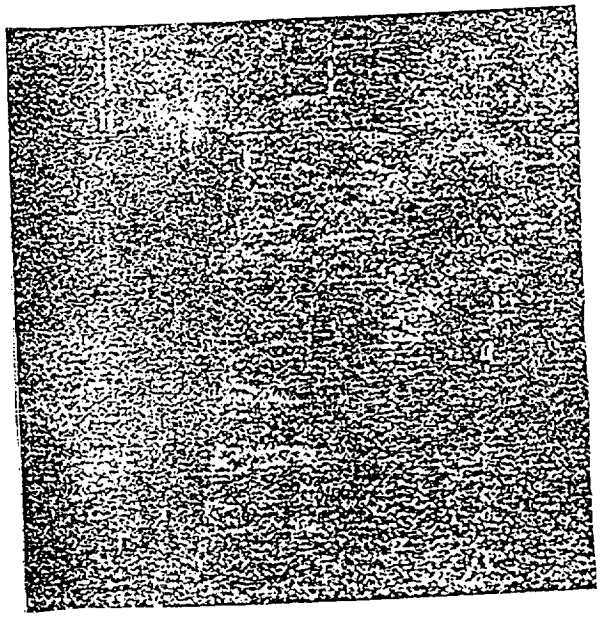
A



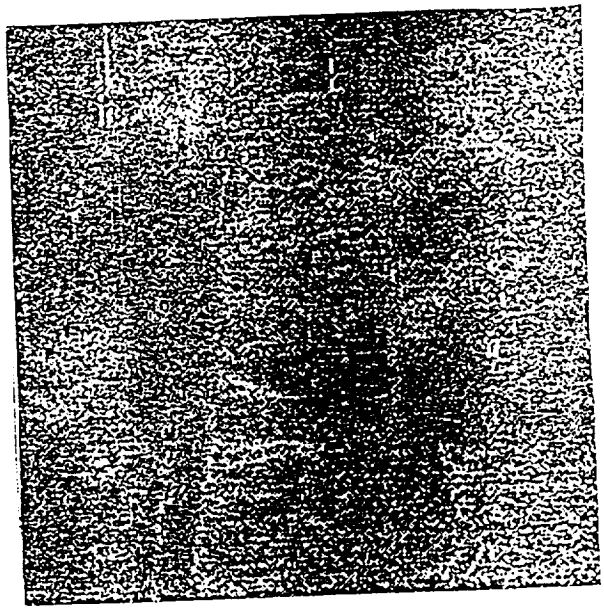
B



C



D



E

Figure 4. Error image for couple image at
a) rate=2.0, b) rate=2.08, c) rate=2.13, d) rate=2.23 e) rate=2.24

Figure 7. Lena image coded at two bits per pixel using DCT with fixed bit allocation

Figure 8. Lena image coded at (a) two bits per pixel and (b) one bit per pixel using JPEG algorithm

Figure 10. Lena image coded at (a) 2 bits per pixel and (b) 1.35 bits per pixel using EPDPCM

A Joint Source/Channel Coder Design*

Khalid Sayood
Dept. of Electrical Engineering
and the
Center for Communication& Information Science
University of Nebraska
Lincoln, NE 68588-0511
Telephone: (402) 472-6688
FAX: (402) 472-4732
email: ksayood@eecomm.unl.edu

Fuling Liu
Western Atlas Company
Houston, Texas

Jerry D. Gibson
Dept. of Electrical Engineering
Texas A&M University
College Station, TX 77843

COMSOC Technical Committee: Signal Processing and Communication Electronics
Hot topic session number: HT28

Abstract

We examine the situation where there is residual redundancy at the source coder output. We have previously shown that this residual redundancy can be used to provide error correction without a channel encoder. In this paper we extend this approach to conventional source coder/convolutional coder combinations. We also develop a design for nonbinary encoders for this situation. We show through simulation results that the proposed systems consistently outperform conventional source-channel coder pairs with gains of greater than 10dB at high probability of error.

*This work was supported in part by NASA Lewis Research Center (NAG 3-806) and NASA Goddard Space Flight Center (NAG 5-916)

1 Introduction

One of Shannon's many fundamental contributions was his result that source coding and channel coding can be treated separately without any loss of performance for the overall system [1]. The basic design procedure is to select a source encoder which changes the source sequence into a series of independent, equally likely binary digits followed by a channel encoder which accepts binary digits and puts them into a form suitable for reliable transmission over the channel. However, the separation argument no longer holds if either of the following two situations occur:

i. The input to the source decoder is different from the output of the source encoder, which happens when the link between the source encoder and source decoder is no longer error free, or

ii. The source coder output contains redundancy.

Case (i) occurs when the channel coder does not achieve zero error probability and case (ii) occurs when the source encoder is suboptimal. These two situations are common occurrences in practical systems where source or channel models are imperfectly known, complexity is a serious issue, or significant delay is not tolerable. Approaches developed for such situations are usually grouped under the general heading of joint source/channel coding.

Most joint source channel coding approaches can be classified in two main categories; (A) approaches which entail the modification of the source coder/decoder structure to reduce the effect of channel errors, [2-18] and (B) approaches which examine the distribution of bits between the source and channel coders [19-21]. The first set of approaches can be divided still further into two classes. One class of approaches examines the modification of the overall structure [2-10], while the other deals with the modification of the decoding procedure to take advantage of the redundancy in the source coder output.

In this paper we present an approach to joint source/channel coder design, which belongs to category A, and hence we explore a technique for designing joint source/channel coders, rather than ways of distributing bits between source coders and channel coders. We assume that the two nonideal situations referred to earlier are present. For a nonideal source coder, we use MAP arguments to design a decoder which takes advantage of redundancy in the source coder output to perform error correction. We have previously shown that this approach can provide error protection at high error rates [16, 17]. In this paper we show that the use of such a decoder in conjunction with a channel encoder can provide excellent error protection over a wide range of channel error probabilities. We then use the decoder structure to infer a channel encoder structure which is similar to a nonbinary convolutional encoder.

2 The Design Criterion

For a discrete memoryless channel (DMC), let the channel input alphabet be denoted by $A = \{a_0, a_1, \ldots, a_{M-1},\}$, and the channel input and output sequences by $Y = \{y_0, y_1, \ldots, y_{L-1}\}$ and $\hat{Y} = \{\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_{L-1}\}$, respectively. If $\mathcal{A} = \{A_i\}$ is the set of sequences $A_i = \{\alpha_{i,0}, \alpha_{i,1}, \ldots, \alpha_{i,L-1}\}$, $\alpha_{i,k} \epsilon A$, then the optimum receiver (in the sense of maximizing the probability of making a correct decision) maximizes $P[C]$, where

$$P[C] = \sum_{A_i} P[C|\hat{Y}]P[\hat{Y}].$$

This in turn implies that the optimum receiver maximizes $P[C|\hat{Y}]$. When the receiver selects the output to be A_k, then $P[C|\hat{Y}] = P[Y = A_k|\hat{Y}]$. Thus, the optimum receiver selects the sequence A_k such that

$$P[Y = A_k|\hat{Y}] \geq P[Y = A_i|\hat{Y}] \ \forall_i.$$

Noting that

$$P(Y|\hat{Y} = \frac{P(\hat{Y}|Y)P(Y)}{P(\hat{Y})}$$

and for fixed length codes $P(\hat{Y})$ is irrelevant to the receiver's operation, the optimal receiver maximizes $P(\hat{Y}|Y)P(Y)$. If we impose a first order markov assumption on $\{y_i\}$, we can easily show that

$$P\left(\hat{Y}|Y\right)P(Y) = \prod P(\hat{y}_i|y_i)\,P(y_i|y_{i-1})$$

This result addresses the situation in case (ii), i.e., the situation in which the source coder output (which is also the channel input sequence) contains redundancy. Using this result, we can design a decoder which will take advantage of dependence in the channel input sequence. The physical structure of the decoder can be easily obtained by examining the quantity to be maximized. The optimum decoder maximizes $P(\hat{Y}|Y)P(Y)$ or equivalently $\log P(\hat{Y}|Y)P(Y)$, but

$$\log P(\hat{Y}|Y)P(Y) = \sum \log P(\hat{y}_i|y_i)P(y_i|y_{i-1}) \tag{1}$$

which is similar in form to the path metric of a convolutional decoder. Error correction using convolutional codes is made possible by explicitly limiting the possible codeword to codeword transitions, based on the previous code input and the coder structure. At the receiver the decoder compares the received data stream to the *a priori* information about the code structure. The output of the decoder is the sequence that is most likely to be the transmitted sequence. In the case where there is residual strucure in the source coder output, the structure makes some sequences more likely to be the transmitted sequence, given a particular received sequence. In other words, even when there is no structure being imposed by the encoder, there is sufficient residual structure in the source coder output that can be used for error correction. The structure is reflected in the conditional probabilities, and can be utilized via the path metric in (1) in a decoder similar in

structure to a convolutional decoder. However, to implement this decoder we need to be able to compute the path metric.

Examining the branch metric, we see that it consists of two terms $\log P(\hat{y}_i|y_i)$ and $\log P(y_i|y_{i-1})$. The first term depends strictly on our knowledge of the channel. The second term depends only on the statistics of the source sequence. In our simulation results we have assumed that the channel is a binary symmetric channel with known probability of error. We have obtained the second term using a training sequence.

In [17] we showed that the use of the decoder led to dramatic improvements under high error rate conditions. However at low error rates the performance improvement was from nonexistent to minimal. This is in contrast to standard error correcting approaches, in which the greatest performance improvements are at low error rates, with a rapid deterioration in performance at high error rates. In this work we combine the two approaches to develop a joint source channel codec which provides protection equal to the standard channel encoders at low error rates while also providing significant error protection at high error rates.

3 Convolutional Encoders and Joint Source/Channel Decoder

As convolutional coders provide excellent error protection at low error rates, and have a decoder structure similar to the JSC decoder, one way we can combine the two approaches is to obtain the transition probabilities of the convolutional encoder output and use the Joint Source/Channel (JSC) decoder described above instead of the conventional convolutional decoder.

The convolutional decoder uses the structure imposed by the encoder and the Hamming metric to provide error protection. The decoder does not use any of the residual structure from the source coder output. We can make use of the residual structure by noting that the path labels transmitted by the convolutional encoder comprise the channel input alphabet $\{y_i\}$. We can then use a training sequence to obtain the transition probabilities $P(y_i|y_{i-1})$, and an estimate of the channel error

probability to obtain $P(\hat{y}_i|y_i)$. These can be used to compute the branch metric L which can be used instead of the Hamming metric in the decoder.

We simulated this approach using a two bit DPCM system as the source encoder. We used the two images shown in Figure 1 as the source. The USC Girl image was used for training (obtaining the requisite transition probabilities) and the USC Couple image for testing. The output of the DPCM system was encoded using a (2,1,3) convolutional encoder with connection vectors

$$g^{(1)} = 64 \quad g^{(2)} = 74.$$

The convolutional encoder was obtained from [23]. The performance of the different systems was evaluated using two different measures. One was the reconstruction signal-to-noise ratio (RSNR) defined as

$$RSNR = 10\log_{10}\frac{\sum u_i^2}{\sum (u_i - \hat{u}_i)^2}$$

where u_i is the input to the source coder (source image) and $\hat{u}_i$ is the output of the source decoder (reconstructed image). The other performance measure was the decoded error probability. The received sequence was decoded using either a standard convolutional decoder or the JSC decoder. A block diagram of the system is shown in Figure 2. The results are presented in Figure 3. While there is some improvement in the decoded error probability for high error rates, the RSNR actually goes down for the MAP decoded sequence. This is somewhat disappointing until one realizes that the JSC decoder makes use of the structure in the nonbinary output of the source coder. When we used the (2,1,3) coder we destroyed some of this structure because the source coder puts out two bit words while the channel coder codes the input one bit at a time. Therefore, if we could preserve the structure in the source coder output by coding the two bit words as a unit we should get improved performance. To verify this we conducted another set of simulation with a rate 1/2

(4,2,1) convolutional code with connection vectors

$$g_1^{(1)} = 6 \quad g_1^{(2)} = 0 \quad g_1^{(3)} = 6 \quad g_1^4 = 4$$
$$g_2^{(1)} = 0 \quad g_2^{(2)} = 6 \quad g_2^{(3)} = 4 \quad g_2^{(4)} = 2.$$

In this case there is a one-to-one match between the source coder output and the channel coder input, and the results shown in Figure 4 reflect this fact. There is considerable improvement in the decoded error probability and there is about a 5 dB improvement obtained by using the MAP decoder at a probability of error of 0.1. These results justify the contention that for best use of the JSC decoder the input alphabet size of the channel coder should be the same as the size of the output alphabet of the source coder. To this point we have been using a MAP decoder with an encoder designed to maximize performance with a Hamming metric. In the next section we propose a general channel coder design to go with the map decoder which has the added flexibility of being able to match the size of the source coder output alphabet.

4 A Modified Convolutional Encoder

Given that the preservation of the structure in the source coder output requires the channel coder input alphabet to have a one-to-one match with the generally nonbinary source coder, we propose a general nonbinary convolutional encoder (NCE) whose input alphabet has the requisite property.

Let x_n, the input to the NCE, be selected from the alphabet $A = \{0, 1, 2, ..., N-1\}$, and let y_n, the output alphabet of the NCE, be selected from the alphabet $S = \{0, 1, 2, ..., M-1\}$. Then the proposed NCEs can be described by the following mappings

Rate 1/2 NCE: $M = N^2$; $y_n = Nx_{n-1} + x_n$

Rate 1/3 NCE: $M = N^3$; $y_n = N^2x_{n-2} + Nx_{n-1} + x_n$

Rate 2/3 NCE: $M = N^3$; $y_n = N^2x_{2n-2} + Nx_{2n-1} + x_2n$

Because of lack of space we will only describe and present the results for the rate 1/2 NCE. The description and results for the other cases can be found in [24] and are similar to the results for the rate 1/2 NCE code.

The number of bits required to represent the output alphabet of the NCE codes using a fixed length code is

$$\lceil \log_2(M) \rceil = \lceil \log_2(N^2) \rceil = \lceil 2\log_2(N) \rceil$$

Therefore in terms of rate, the rate 1/2 NCE coder is equivalent to a rate 1/2 convolutional encoder. The encoder memory in bits is $2\lceil \log_2(N) \rceil$ as each output value depends on two input values.

As an example, consider the situation when $N = 4$. Then $A = \{0,1,2,3\}$ and $S = \{0,1,2,...,15\}$. Given the input sequence x_n : 0 1 3 0 2 1 1 0 3 3 and assuming the encoder is initialized with zeros, the output sequence will be y_n : 0 1 7 12 2 9 5 4 3 15.

The encoder memory is four bits. Notice that while the encoder output alphabet is of size N^2, at any given instant the encoder can only emit one of N different symbols as should be the case for a rate 1/2 convolutional encoder. For example if $y_{n-1} = 0$, then y_n will take on a value from $\{0,1,2,...,(N-1)\}$. In general, given a value for y_{n-1}, y_n will take on a value from $\{\alpha N, \alpha N+1, \alpha N+2, ..., \alpha N+N-1\}$, where $\alpha = y_{n-1}(mod N)$. This structure can be used by the decoder to provide error protection. The encoder is shown in Figure 5.

4.1 Binary Encoding of the NCE Output

We will make use of the residual structure in the source coder output (which is preserved in the NCE output) at the receiver. However, we can also make use of this structure in selecting binary codes for the NCE output. An intelligent assignment of binary codes can improve the error correcting performance of the system.

When each allowable sequence is equally likely, there is little reason to prefer one particular

assignment over others. However, when certain sequences are more likely to occur than others, it would be useful to make assignments which increase the 'distance' between likely sequences. While, for small alphabets it is a simple matter to assign the optimum binary codewords by inspection, this becomes computationally impossible for larger alphabets. We use a rather simple heuristic which, while not optimal, provides good results.

Our strategy is to try to maximize the Hamming distance between codewords that are likely to be mistaken for one another. First we obtain a partition of the alphabet based on the fact that given a particular value for y_{n-1}, y_n can only take on values from a subset of the full alphabet. To see this, consider the rate 1/2 NCE; then the alphabet S can be partitioned into the following sub-alphabets:

$$S_0 = (0, 1, 2, 3..., N-1)$$

$$S_1 = (N, N+1, ..., 2N-1)$$

$$\vdots$$

$$S_{N-1} = \left(N(N-1), N(N-1)+1, ..., N^2-1\right)$$

where the encoder will select letters from alphabet S_j at time n if $j = y_{n-1} (\mathrm{mod} N)$. Now for each sub-alphabet we have to pick N codewords out of M $(= N^2)$ possible choices. We first pick the sub-alphabet containing the most likely letter. The letters in the sub-alphabet are ordered according to their probability of occurrence. We assign a codeword a from the list of available codewords to the most probable symbol. Then, assign the complement of a to the next symbol on the list. Therefore the distance between the two most likely symbols in the list is $K = \lceil \log_2 M \rceil$ bits. We then pick a codeword b from the list which is at a Hamming distance of $K/2$ from a and assign it and its complement to the next two elements on the list. This process is continued with the selection of letters that are$K/2^k$ away from a at the k^{th} step until all letters in the subalphabet have a codeword assigned to them.

4.2 Simulation Results

The proposed approach was simulated using the same setup as was used in the preceding simulations. We show the results for the rate 1/2 NCE coder in Figure 6 and comparisons in Figure 7. Note the dramatic improvement in performance with the rate 1/2 NCE code. At a probability of error of 0.1 the RSNR drops by less than 1 dB. For the same channel conditions the use of the (2,1,3) code results in a drop of more than 10 dB. Looking at the decoded error probabilities, even when the channel error probability is 0.25, the decoded error probability is less than 10^{-2}. This improvement has been obtained with only a minimal increase in complexity. Similar results have also been obtained for rate 1/3 and 2/3 NCE codes.

5 Conclusion

If the source and channel coder are designed in a "joint" manner, that is the design of each takes into account the overall conditions (source as well as channel statistics), we can obtain excellent performance over a wide range of channel conditions. In this paper we have presented one such design. The resulting performance improvement seems to validate this approach, with a "flattening out" of the performance curves. This flattening out of the performance curves makes the approach useful for a large variety of channel error conditions.

References

[1] C. E. Shannon, *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[2] J. G. Dunham and R. M. Gray, *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 516–519, July 1981.

[3] E. Ayanoğlu and R. M. Gray, *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 855–865, Nov. 1987.

[4] J. G. Dunham and R. M. Gray, *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 516–519, July 1981.

[5] J. L. Massey, in *Communication Systems and Random Process Theory*, J. K. Skwirzynski, ed., Sijthoff and Nordhoff: Netherlands, pp. 279–293, 1978.

[6] T. C. Ancheta, Jr. Ph.D. dissertation, Dept. of Electrical Engr., Univ. of Notre Dame. Aug. 1977.

[7] K-Y Chang and R. W. Donaldson, *IEEE Trans. Commun.*, vol. COM-20, pp. 338–350, June 1972.

[8] A. J. Kurtenbach and P. A. Wintz, *IEEE Trans. Commun. Technol.*, vol. COM-17, pp. 291–302, April 1969.

[9] N. Farvardin and V. Vaishampayan, *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 827–838, November 1987.

[10] D. J. Goodman and C. E. Sundberg, *Bell Syst. Tech. J.*, vol. 62, pp. 2017–2036, Sept. 1983.

[11] D. J. Goodman and C. E. Sundberg, *Bell Syst. Tech. J.*, vol. 62, pp. 2735–2764, Nov. 1983.

[12] R. C. Reininger and J. D. Gibson, *IEEE Trans. Commun.*, vol. COM-31, pp. 572–577, April 1983.

[13] R. Steele, D. J. Goodman, and C. A. McGonegal, *IEEE Trans. Commun.*, vol. COM-27, pp. 252–255, January 1979.

[14] R. Steele, D. J. Goodman, and C. A. McGonegal, *Elec. Lett.*, vol. 13, pp. 351–353, June 1977.

[15] K. N. Ngan and R. Steele, *IEEE Trans. Commun.*, vol. COM-30, pp. 257–269, January 1982.

[16] K. Sayood and J. C. Borkenhagen, *Proceedings IEEE International Conference on Communications*, June 1986, pp. 1888–1892.

[17] K. Sayood and J. C. Borkenhagen, *IEEE Transactions on Communications*, vol. COM-39, June 1991.

[18] K. Sayood and J. D. Gibson, *Proc. 22nd Annual Conference on Information Sciences and Systems*, Princeton, NJ, Mar. 1988, pp. 380-385.

[19] J. W. Modestino, D. G. Daut, and A. L. Vickers, *IEEE Trans. Commun.*, vol. COM-29, pp. 1262–1274, Sept. 1981.

[20] D. Comstock and J. D. Gibson, *IEEE Trans. Commun.*, vol. COM-32, pp. 856–861, July 1984.

[21] C. C. Moore and J. D. Gibson, *IEEE Trans. Commun.*, vol. COM-32, August 1984.

[22] K. Sayood, J. D. Gibson, and F. Liu, *Proc. 22nd Annual Asilomar Conference on Circuits, Systems, and Computers*, Nov. 1988, pp. 102-106.

[23] S. Lin and D. J. Costello, *Error Control Coding*, Prentice Hall, 1983.

[24] K. Sayood, F. Liu and J. D. Gibson, *IEEE Trans. Commun.* To be submitted.

Table 1: **Codeword Assignments**

Symbol	Code	Symbol	Code
0	0000	8	1011
1	0011	9	0111
2	1100	10	0100
3	1111	11	1000
4	1110	12	0101
5	1101	13	1001
6	0001	14	1010
7	0010	15	0110

Figure 1. Images used in simulation

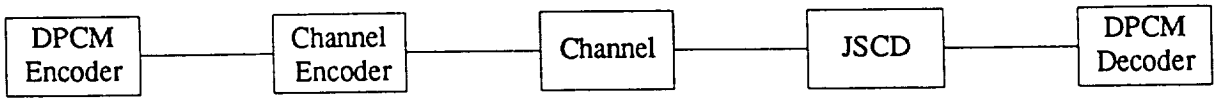


Figure 2. Block Diagram of Proposed System

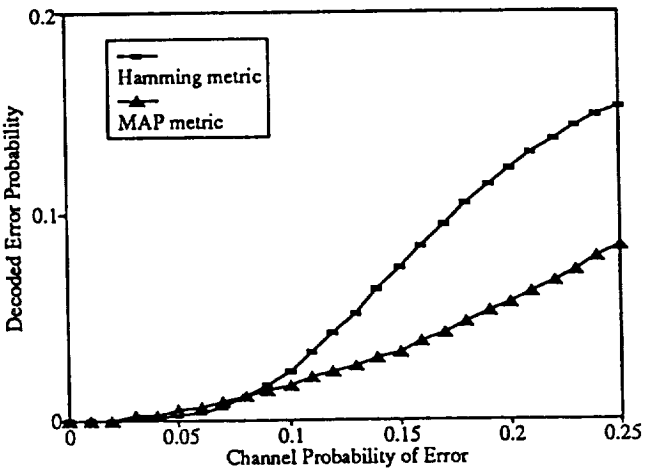


Figure 3a. Decoded error probability for the (2,1,3) convolutional coder.

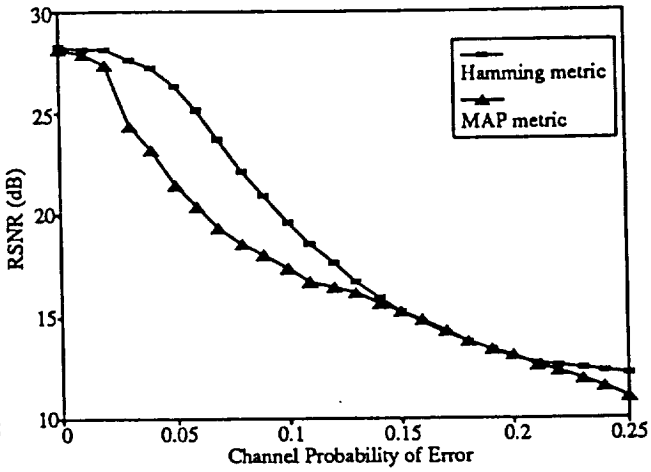


Figure 3b. RSNR for the (2,1,3) convolutional coder

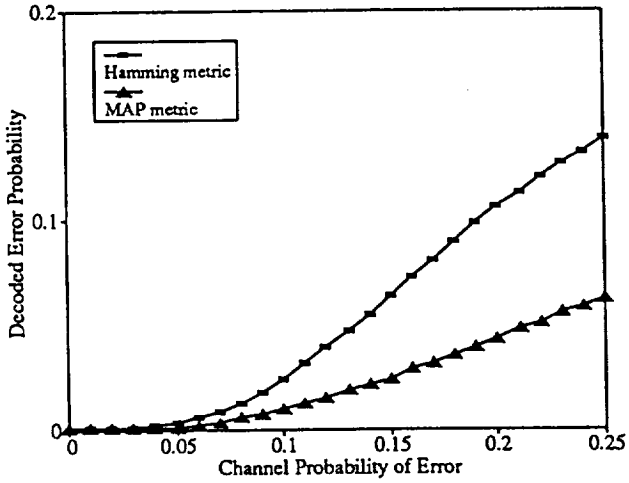


Figure 4a. Decoded error probability for the (4,2,1) convolutional coder

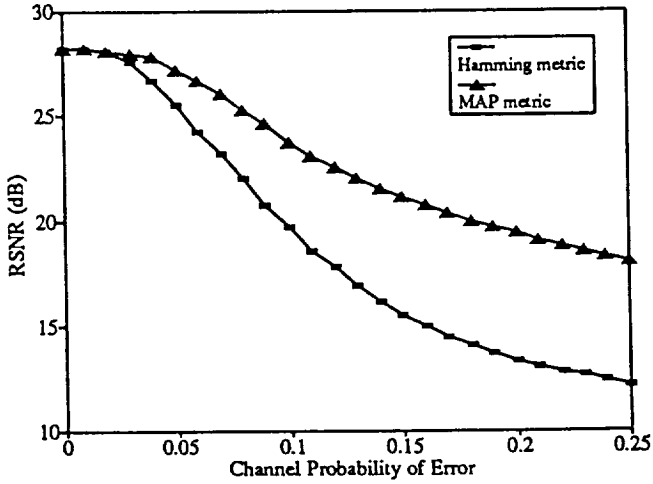


Figure 4b. RSNR for the (4,2,1) convolutional coder

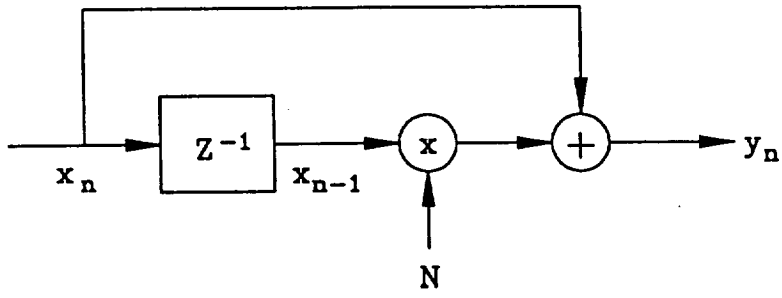


Figure 5. Rate 1/2 Nonbinary Convolutional Encoder

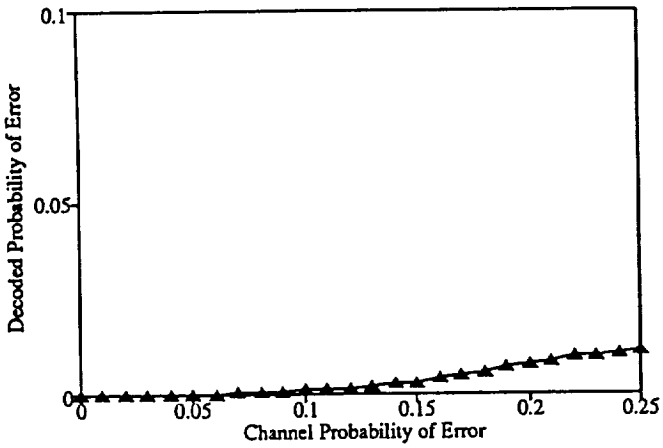


Figure 6a. Decoded vs channel error for rate 1/2 NCE

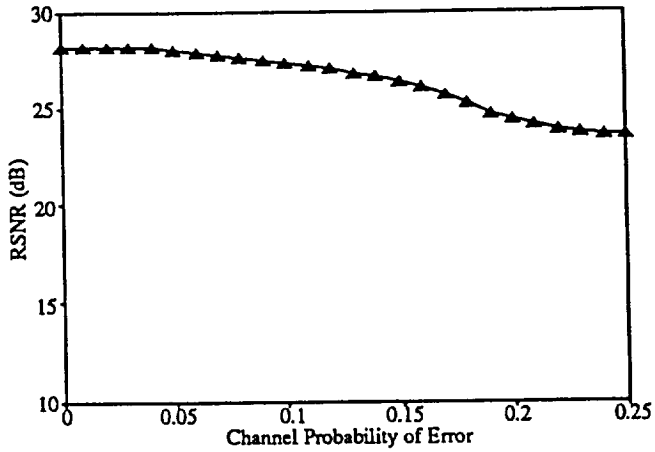


Figure 6b. RSNR for rate 1/2 NCE coder vs channel error

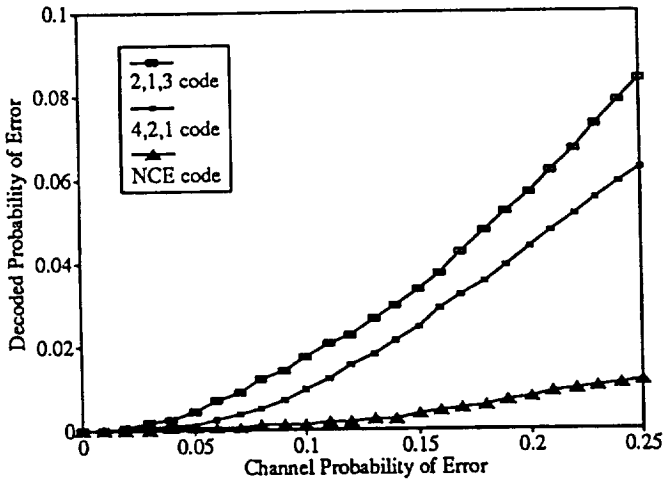


Figure 7a. Decoder error probability for the three MAP decoded systems

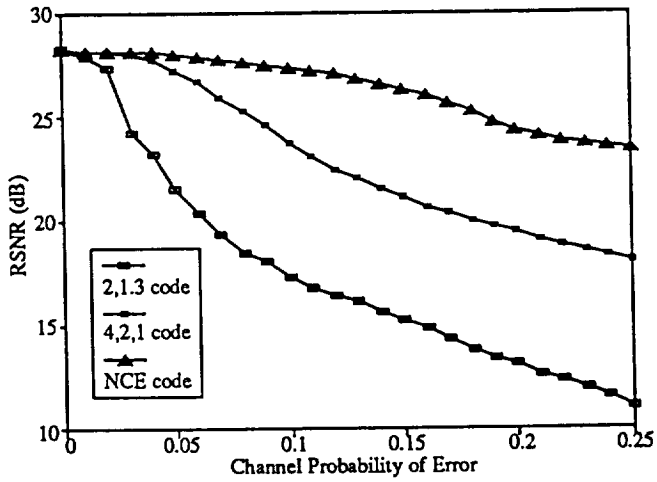


Figure 7b. RSNR vs channel error for the three MAP decoded systems