CR-166225

STUDY OF COMMUNICATIONS

DATA COMPRESSION METHODS

INTERIM
FINAL REPORT

Contract NAS 2-9703

H.W. JONES

July, 1978

(NASA-CR-166225) STUDY OF COMMUNICATIONS DATA COMPRESSION METHODS Interim Final Report (COM-CODE, Inc., Mountain View, Calif.) 267 p HC A12/MF A01 CSCL 17B

N83-17774

Unclas
G3/32 32908

Prepared for
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA, 94035

NASA TECHNICAL MONITOR
Larry B. Hofman

COM-CODE, INC.
305 EASY ST. NO.9 MTN. VIEW
CALIFORNIA 94043

CONTENTS

SECTION A

SUMMARY REPORT

This section provides a brief summary report of the work accomplished under the "Study of Communications Data Compression Methods", under NASA contract NAS 2-9703. The results are fully explained in subsequent sections on video compression, Landsat image processing, and satellite communications.

The first task of contract NAS 2-9703 was to extend a simple monochrome conditional replenishment system to higher compression and to higher motion levels, by incorporating spatially adaptive quantizers and field repeating. Conditional replenishment combines intraframe and interframe compression, and both areas were to be investigated. The gain of conditional replenishment depends on the fraction of the image changing, since only changed parts of the image need to be transmitted. If the transmission rate is set so that only one-fourth of the image can be transmitted in each field, greater change fractions will overload the system.

To accomplish task I, a computer simulation was prepared which incorporated 1) field repeat of changes, 2) a variable change threshold, 3) frame repeat for high change, and 4) two mode, variable rate Hadamard intraframe quantizers. The field repeat gives 2:1 compression in moving areas without noticeable degradation. Variable change threshold allows some flexibility in dealing with varying change rates, but the threshold variation must be limited for acceptable performance. 2:1 frame repeat provides 15 frames per second, and is acceptable, but 4:1 repeat is objectionably degraded. The two mode Hadamard quantizers use a 2 bit per picture element (bpp) edge subpicture quantizer and a 1 bpp flat subpicture quantizer. For the Reasoner image, the two mode variable rate performance at 1.25 bpp is comparible to single mode performance at 2 bpp, but the gains depend on the fraction of changed blocks using each mode.

The goal of task I was to achieve performance free of artifacts at 1/4 bpp. The simulated system achieves this, for scenes with less than 50 percent motion. The different compression ratios for changed parts of the image are 2:1 for conditional replenishment, 2:1 for field repeat, 2:1 for frame repeat, and 4:1 for intraframe Hadamard compression (all edge mode). The total combined compression ratio is 32:1, which reduces the original 8 bpp to 1/4 bpp. If the fraction changing exceeds 50 percent, additional frame repeats occur, with a limit of 4:1 for 100 percent change.

Task II of the statement of work was to extend conditional replenishment to color video. This was accomplished by modifying the monochrome simulation program to process both monochrome and color. The I and Q color signals are quantized using 1/4 bpp in the edge mode, and 1/8 bpp in the flat mode, so that color transmission requires 25 percent higher transmission rate than monochrome. Changes and modes are detected using only the I or monochrome signal, which introduces some artifacts.

Certain related aspects of video compression were investigated in conjunction with these tasks. Frame interlace is a compression method which has some aspects of intraframe compression, and some aspects of interframe compression. Using previously available data, a report was prepared to compare the hardware requirements and performance of field and frame compression. It was concluded that the frame interlace gains about 1/2 bpp in compression, at a cost of 4 or 7 bit-frames of memory, depending on implementation.

D. Hein and N. Ahmed, under a grant from NASA-ARC, recently developed a method for deriving the discrete cosine transform from the Hadamard transform. This method is being implemented at NASA-ARC. In cooperation with D. Hein, transforms intermediate between the discrete cosine and Hadamard transform in complexity and performance were devised and investigated. Theoretical and experimental results show that transforms with simplified implementation can provide part of the gains of the discrete cosine transform over the Hadamard.

Monochrome and color conditional replenishment using several frame repeats was processed using frame averaging. Repeated frames were replaced by composite frames, formed by averaging the two transmitted frames closest in sequence. Frame averaging makes a small improvement in material using three frame repeats, or 10 frames per second, but introduces strange blurring and shape changing in material using five repeats, or 6 frames per second.

Task III of the contract was to investigate image compression for the transmission of Landsat images, with emphasis on using spectral correlation. The proposal noted that cluster coding was a sucessful compression method using spectral correlation, but had high complexity. A new method, "picture element replication", was simulated. A table of previously transmitted elements is maintained at the transmitter and receiver . When the image element to be transmitted is within

a small error distance of an element in the table, the location of the table element is transmitted, instead of the element value. The table element replaces the original element in the image. When no table element is sufficiently similar to the element to be transmitted, the exact element value is transmitted, and inserted in the table. Transmission rate is reduced because the number of table elements is much smaller the number of all possible elements.

Picture element replication was analized theoretically. The expected error is a known fraction of the error bound used to select table elements. If the table size is sufficiently large, the transmission rate is only one or two bpp more than required to transmit a table entry. Increasing table size reduces error and increases rate in a known way. Additional compression can be gained by using a one bit word for a replicated element.

Picture element replication was simulated, and tested with two images used in other work done for NASA-ARC. The new method was found superior to the previous spatial compression methods, and comparable to cluster coding. Picture element replication is much simpler in implementation than cluster coding, and is suitable for the transmission of stored data via satellite to a remote site.

In conjunction with this task, a survey of Landsat classification methods was made. There are several potentially useful techniques reported in the literature. Methods of nonparametric classification, including nearest neighbor, can reduce the human intervention required by normal distribution, maximum liklihood classifiers. Spatial-spectral classifiers can produce classified images similar to those produced by human image interpreters.

Task IV of the statement of work was to investigate signal processing, channel geometry, and demand assignment for satellite communication. The proposal stated the intention to review operational systems, and to evaluate a technique to improve the bandwidth efficiency of satellite communications. The technique selected for study, which is mentioned in the recent literature, is four dimensional signal design. The report on four dimensional bandwidth efficient modulation reviews the advantages of higher dimensional signal design, compared to conventional multi-level amplitude phase shift keying. The potential gain of four dimensional signal design is estimated using the capacity bound.

Two new classes of signal designs are introduced, which counter the disadvantages of amplitude phase shift keying. One class of designs is based on the densest four dimensional lattice, and is conjectured to approach the optimum designs. At transmission rates needed for bandwidth efficient modulation, 2 to 3 bits per dimension, the new designs increase rate 0.6 bits per dimension at fixed signal to noise, or allow 2 to 4 dB signal reduction at fixed rate. The four dimensional receivers are not much more complex than conventional receivers.

In the review of systems, the TRW "Mobile Multiple Access Study" was examined. The study observed that continuously variable slope delta modulation was nearly competitive with frequency modulation (FM). Several other papers on satellite communication, and several power budgets, were reviewed. It seems generally accepted that frequency division multiple access (FDMA) provides more channels in a given bandwidth than time division or code division. Demand assignment of channels by a central station can achieve 100 percent useage, while contention systems are limited to lower efficiency. Of special interest were two Comsat papers. Campanella and others concluded that digital voice modulation and FM have similar performance for single channel per carrier voice channels. Welti and Kwan also found digital competitive, and considered some four dimensional methods in their comparison.

Some results of this study will be published. The operation of an earlier conditional replenishment system was described in two papers given at the August 1977 SPIE convention and at the december 1977 NTC meeting. The study of frame and field compression, appendix A of section B below, will be presented at SPIE in August 1977. The investigation of transforms derived via the Hadamard transform, appendix B of section B below, has been accepted for the ITC meeting in November 1978. The report on Landsat element replication compression, section C below, will be published in the IEEE Transactions on Communications if certain required revisions are made.

SECTION B

VIDEO COMPRESSION

MONOCHROME AND COLOR

CONDITIONAL REPLENISHMENT

CONTENTS

This report describes the newly developed conditional replenishment system. The conditional replenishment simulation program has been modified to have the following features:

1) forced field repeat of changed subpictures
2) a range of thresholds differing by one for all vectors tested; the threshold used is the lowest that does not give excessive changes
3) frame repeat, when the number of changes is excessive and the change detection threshold is at its upper limit
4) two mode replacement of changed Hadamard subpictures, using optimized flat and edge quantizers with fixed rate
5) a forced refresh option, in addition to refresh using any rate not needed for update of changed subpictures.

The forced field repeat of changes, with rate buffering over a frame, reduces the transmission rate required for changes by 2:1. This causes no apparent loss of motion rendition quality, but there is a reduction in vertical resolution which is acceptable in moving areas. Selecting the change detection threshold based on the number of changes for the current frame, rather than for past numbers of changes, eliminates threshold caused high change problems and is needed to allow the use of optimized fixed rate quantizers. Frame repeat is sometimes better during high changes than replacing only the larger changes by increasing the change detection threshold, which causes a panned scene to break up into subpictures. Using optimized quantizers and a two mode intraframe compression reduces the average rate required to replace a changed subpicture, as further discussed below. The refresh is used both to repair transmission errors and to bring unchanging subpictures to the full 3 bpp resolution in both video fields.

The compression ratios of the various techniques used in replacing changed subpictures are shown in Table I, for operation at an average rate of 3/16 bpp over a frame. The average rate required to replace a changed subpicture using two mode compression is estimated to be 1 1/2 bpp. With forced field repeat, the configuration can operate at fractions of change up to 1/4 without frame repeat. In the case of a complete change, the frame rate drops to 1/4, or 7.5 frames per second. Forced

Table I: Compression techniques and ratios for change replacement.

Method	Compression ratios			
1. Intraframe, two mode	8:1 1/2 (or 5 1/3:1)			
2. Conditional replenishment	4:1	2:1	4/3:1	1:1
3. Forced field repeat	2:1			
4. Frame repeat, as needed	1:1	2:1	3:1	4:1
Total compression ratio	42 2/3 :1			
Acceptable change fraction	<1/4	<1/2	<3/4	≤ 1
Frame rate, per second	30	15	10	7 1/2

refresh could require another 1/16 bpp, for complete replacement of both fields in 48 frames or 1 1/2 seconds. The change indicator overhead is 1 bit for each subpicture, or 1/64 bpp. The flat/edge mode indicator overhead is another 1 bit for each changed subpicture, or 1/64 bpp which can be included in the 1 1/2 bpp for changed subpictures. The total average rate is 3/16 + 1/16 + 1/64, or 17/64 bpp, for monochrome.

A block diagram of the simulated system is given in figure 1. All memories are at 3 bpp, and use the organization of the 3 bpp list quantizations. The 3 bpp quantization is obtained by rounding and truncating the transform coefficients, without using read-only memories. The receiver memories often actually contain lower quality descriptions of the subpictures, because of the use of lower rate quantizations based on read-only memories for changed subpicture replacement. The system can probably be implemented with a two frame delay, exclusive of repeats. During the first frame time, the first field of the frame is transformed truncated, then change detected and flat/edge mode detected, and the change information placed in the transmitter buffer. During the second frame time, the change and refresh information is transmitted to the receiver buffer. During the third frame time, the receiver buffer is used to update the receiver memory and to form the display, in conjunction with the memory. In the repeat mode, the display formed entirely from memory, and the change information data is stored in the repeat memory until the new frame is complete.

We further consider the optimized, fixed rate quantizations used to replace changed subpictures. The previous system used a high quality, 3 bpp, quantization for the transformed vector memory, as does the revised system. The quantization is divided into twenty-four lists of vectors, each list at 1/8 bpp, for use in variable rate change replacement, memory completion, and refresh. While the revised system retains the 3 bpp quantization and lists for memory organization and refresh, the changed subpictures are replaced using two-mode, optimized, fixed rate quantizers, and there is no memory completion other than that provided by the refresh.

The purpose of two mode, optimized, fixed rate quantizers is to reduce the rate

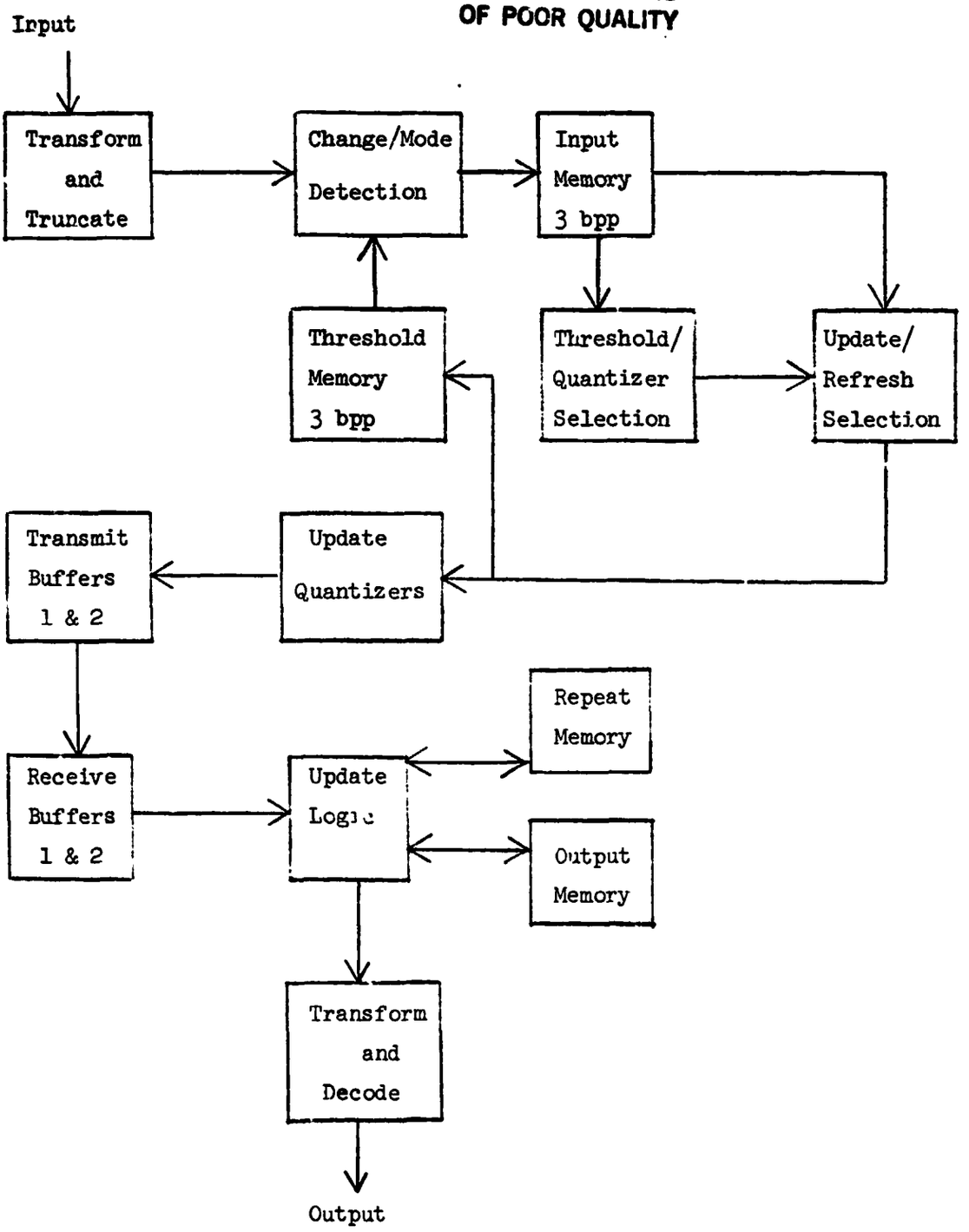


Figure 1: Conditional replenishment system block diagram.

required to replace a changed subpicture, while retaining the required quality. Figure 2 shows the mean square error for various intra frame quantizers, for the Reasoner image processed in the field. The previous system attempted to use at least sixteen lists, or 2 bpp, to replace changed subpictures. The earlier and improved fixed rate, single mode quantizers for 1 1/2 bpp have about the same mean square error as the list quantizer at 2 bpp. By using fixed rate, single mode quantizers, a gain of 1/2 bpp can be made. The penalty for using quantizers which are not based on the memory quantization lists, is that completing changed subpictures becomes much more complicated. This memory completion can be dispensed with, if the initial replacement quality is reasonably high, and if the refresh rate is sufficient to supply full quality in a reasonable time.

A further reduction in the rate to replace a changed subpicture can be made using two mode, variable rate intraframe compression. A variable rate for replaced subpictures can easily be used, because a conditional replenishment system allocates rate over a frame. Mean square results for three two-mode, variable rate methods are also given in figure 2. Two mode, variable rate quantization provides significant additional rate reduction. The data point at about 1.1 bpp uses a 1 1/2 bpp edge mode and a 1 bpp flat mode. The points at about 1 1/4 bpp use a 2 bpp edge mode with the same 1 bpp flat mode, and have two different change detection thresholds. The average rate required to replace changed subpictures depends on the amount of detail in moving or changing areas.

Optimizing a two mode, variable rate system is an interesting problem. Assume that we first design an acceptable quantization at some fixed rate. Because large, noticeable errors usually occur at edges, a good subjective quantization is usually optimized for edges. Many subpictures are flat and featureless, and could be transmitted using fewer bits. A flat subpicture quantization would have narrow vector ranges and small quantization steps. In a two mode system , the original acceptable quantization is used for edges, and whenever a vector extends beyond the flat quantization's range. The best two mode design gives the largest rate reduction from the original single mode design. The rate reduction equals the fraction of subpictures using the flat mode, times the difference between edge and flat mode rates. As the

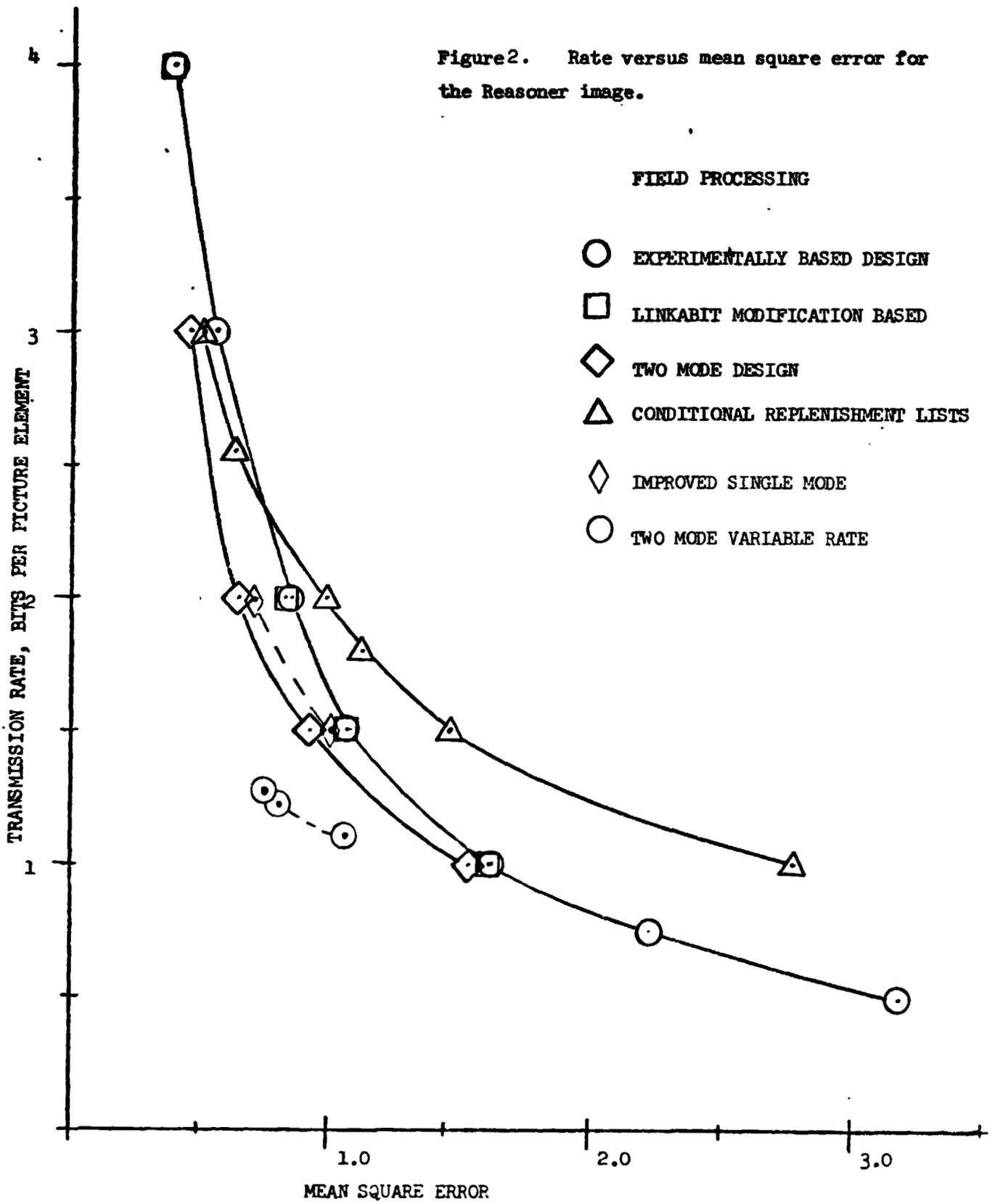


Figure 2. Rate versus mean square error for the Reasoner image.

rate of the flat mode is increased, its rate reduction is reduced, but its vector ranges increase to cover a larger fraction of the subpictures.. Because of these compensating factors, the actual choice is not too critical. A 3/4 bpp flat mode was used for half the subpictures, for a rate reduction of 1/2(2-3/4) = 5/8. A 1 bpp flat design handled 3/4 of the subpictures, for a reduction of 3/4(2-1) = 3/4. The 1 bpp flat design also had superior visual quality, and gave a more intuitively acceptable measure of the portion of flat area in the Reasoner image.

The DCT and two new transforms for intraframe transform compression are described in Appendix B. Figure 8 of Appendix B gives single mode results, also for the Reasoner image, for the discrete cosine and $\tilde{B}$ and $\tilde{C}$ matrix transforms using the experimentally based quantizers designed for the Hadamard transform. Some of the results are compared in Table 2 below, for a mean-square error of 1.0.

TABLE 2

METHOD	RATE FOR MSE =1.0
Hadamard, conditional rep lists	2.0
" single mode	1.6
" improved single mode	1.5
Cosine, single mode, hadamard design	1.4
Hadamard, two mode, fixed rate	1.4
" " variable rate	1.15

The total range of transmission rates is 2.0:1.15 or 1.74:1. If the conditional replenishment lists, which are now used only for memory update, are eliminated, the range of rates is 1.6:1.15 or 1.39:1. The cosine gains 0.2 bits per pel, even using a Hadamard quantizer design, and another 0.1 or 0.2 should be obtainable by a design for the cosine.

In the monochrome conditional replenishment test runs, the Hadamard transform was used with 3 bpp memory quantization, 2 bpp edge mode, and 1 bpp flat mode quantization. A tabulation of all the simulations is given in Table 3, Appendix D.

Conditional replenishment test runs have been made at 1/4, 1/8, and 1/16 bits per sample, although only a few have been viewed on the Echo Science disc. 3/16 bits per pel seems a reasonable rate, as outlined in table I above, for monochrome conditional replenishment. The intraframe rate is taken as 1.5 bpp, for a flat (1 bpp) and an edge (2 bpp) mode used equally. As the fraction changing increases beyond 1/4, frame repeat is introduced. The same table can be used to show how operation at 1/16 bpp is possible. Using 5 1/3:1 for intraframe, 4:1 for conditional replenishment, 2:1 for field repeat, a frame repeat of 4:1 is required. This gives 7 1/2 frames per second, noticably degraded motion. The overall compression ratio is 120:1.

Color intraframe compression was examined as part of an earlier contract, and a report was made in the winter of 1976-7. It was noted that the chrominace signals, IQ or R-Y, B-Y, are usually filtered to 0.5 MHz, while the luminance or Y component has a bandwidth of 4.5 MHz. This implies that the horizontal sampling rate can be reduced 9:1, if suitable pre and post filtering is used. The report suggested that one I or Q color sample be transmitted in each 4 by 4 sample block using 8 bits, for an average color rate of 0.5 bpp. Published work in intraframe transform compression using the I and Q signals indicated that the color components could be transmitted at less thna 1 bpp. Pratt used 0.75 bpp for both color components, and Chen used 0.53 bpp for I and 0.29 bpp for Q.

Experimental work in color compression at ARC was reported by Jones on Aug.15, 1977. Hein, working in the frame, combined transformed but uncompressed Y (8 bpp) with transformed, compressed I and Q. The I and Q signals had the Hadamard H_{00}, H_{01}, H_{10}, and H_{11} represented using 8 bits $\begin{matrix} 8 & 8 \\ 8 & 8 \end{matrix}$. Color performance was acceptable. Jones repeated the above quantization in the field, and also used 2 bpp Y compression similar to that described in the monochrome intraframe section above. Color performance was acceptable, but since an 8 by 8 block was used, the total color rate was 1 bpp. The rate was reduced to 1/2 bpp using $\begin{matrix} 8 & 4 \\ 4 & 0 \end{matrix}$ bits for the I and Q vectors. Reduction to a total of 1/4 bpp using $\begin{matrix} 8 & 0 \\ 0 & 0 \end{matrix}$ for I and Q, but the smear of color at edges was unacceptable. The results of this work were gratifying, because they provide a rare instance where experiment confirms theoretical prediction. However, it should be noted that only a few images were tested, and only Hadamard, single mode quantizations were used.

Investigation of color conditional replenishment was performed as part of task II.D format tapes, in RGB, were converted to E, YIQ. Modifications were made to the monochrome conditional replenishment so that color could be run. This first color program made two important choices:

1) changes in time are detected using Y only, and I and Q are updated when Y is updated.

2) the I and Q signals, like the Y, have two update modes, flat and edge.

If these choices are acceptable, the complexity of a color conditional replenishment system is reduced. These choices are in accordance with Limb's idea (Plateau Coding) that color usually changes only when luminance changes.

The two color files available are Wheel of Fortune (59 frames), and Water Skiers (46 frames). These have been run at 1.0, 0.5,and 0.25total average bpp, using the intraframe bits given in table 3 below.

TABLE 3

	RATE, bpp		
	Memory	Flat	Edge
Y	3	1	2
I	1/4	1/8	1/4
Q	1/4	1/8	1/4
Total	3 1/2	1 1/4	2 1/2

The 1/4 bpp quantization for I and Q is $\begin{matrix} 8 & 4 \\ 4 & 0 \end{matrix}$, the 1/8 bpp quantization is $\begin{matrix} 8 & 0 \\ 0 & 0 \end{matrix}$. Wheel and Skiers have average total change percentages of 50% and 65% at 0.5 bpp, where the threshold is high, and average change percentages of 75% and 98% at 1 bpp, where the change threshold is low. These changes are the highest seen so far, and are probably not typical of teleconference material. The changes at 0.5 bpp are nearly all edge mode in Wheel, but are 40%-50% flat in Skiers, so that the flat mode gives useful rate reduction. Table 3, above, shows that the transmission of color changes requires a 25 percent higher rate than monochrome changes.

Because of the adaptive change detection threshold, the number of changes is a function of the fixed average transmission rate. However, for low rates the threshold reaches the limiting value, and the minimum number of changes is detected. The fixed rate is achieved by frame repeats, and the number of changes fluctuates. This occured for Skiers at the rate of 0.25 bpp. Table 4 gives the typical change rates for these files compressed at average rates of 1.0, 0.5, and 0.25 bpp.

Table 4 Typical change percentages

Average rate	1.0	0.5	0.25
Wheel	75 %	50 %	50 %
Skiers	98 %	65 %	65-75 %

These change rates are very large, and not typical of teleconference material, but they provide an interesting worst case test.

At the highest rate, 1.0 bpp, the adaptive threshold drops to the minimum, increasing the number of changes. At 0.5 and 0.25 bpp, the change detector increases to the maximum, least sensitive setting. The bit rates and modes for the Y, I, and Q signals were indicated in the previous report. The number of repeats and the resultant frame rates are given in table 5.

Table 5 Number of repeats/frames per second

Average rate	1.0	0.5	0.25
Wheel	0/30	1/15	2/10
Skiers	0/30	1/15	2-3/10-7.5

The number of repeats and the frame rates are determined by the average rate allowed and the rate needed to describe the detected changes, as computed from the number and mode of the changes. The number of repeats is much higher than for any of the monochrome runs at the same average rate, largelybecause of the higher changes in the two color test sequences.

The compressed test runs of Wheel and Skiers made at 0.5 and 0.25 bpp were converted to RGB format and placed on the Echo Sciences disk. In general, the results were good, and color artifacts due to two mode intraframe and conditional replenishment compression were not detectable in real time.

Several different sorts of artifacts were found, and these are described below.

Color quality. The colors in general were pale. This is also true of the original, and is due to the difficulty of synchronizing the Echo disk when saturated colors are recorded. In Wheel, the girl's face turns slightly greenish when she moves into the shadow of the wheel, This also occurs in the original. There is no reason to suspect that the compression methods affect color quality.

Color modes. The flat and edge modes were selected using only the Y component. This produced a blurring of the purple stripe arround the door in Wheel, where Y levels are similar. The refresh improved the definition rapidly at 0.5 bpp. The edge/flat mode detection can also depend on I and Q.

Color change detection. In the Skiers sequence, parts of a red-orange reflection are momentarily rendered as blue, the cclor of the object formerly occupying the same image area. Both colors have similar Y level, and the change detection error is not apparent in real time. The change detection can also depend on I and Q.

Moving edges. In Wheel, the more noticeably moving part of the girl's jeans has a busy edge. This is a familiar conditional replenishment artifact for moving high contrast edges.

Field repeat. In current simulations, all moving areas are shown using field repeat. This causes a resolution loss most noticeable in the Skiers tow ropes, which are jagged. This is typical of field repeat, and would be removed by the refresh/update in static areas.

Motion artifacts. There is apparent flicker in the real-time presentation of the conditional replenishment material. Some of this is due to the imbalance between sucessive frames, which use different heads. This can be shown when the same frame is rerecorded, as was done at 0.25 bpp. Some of the flicker is due to hum bars recorded in random vertical positions, causing brightness changes at the frame rate. Finally, there is some jerkiness due to the frame repeats, which are more

apparent at 0.25 bpp.

Other artifacts. In Skiers, some of the lettering on the ramp is strangely interlaced, due to a film interlace problem in the original. Also in Skiers, there are some white blocks which appear to anticipate the motion of the ramp, probably due to the same cause.

In summary, the color artifacts observed in this material are not objectionable in real time, and are either present in the original scene or can be removed by testing I and Q for changes and modes.

APPENDIX A

COMPARISON OF VIDEO FRAMES AND FIELDS

FOR TRANSFORM COMPRESSION

INTRODUCTION

This paper describes the results achieved, and the hardware required, for video compression using either fields or interlaced frames. Interlacing the video fields, and the inverse operation, requires substantial digital memory, but achieves a given compressed image quality using a lower transmission rate.

In television transmission, the scene is repeatedly scanned to form a field image of about 256 lines. As shown in figure 1, each field consists of every second line in the full frame, and the alternating fields (represented by solid or dashed lines) are displaced vertically by one line. Two sucessive fields form the full video frame of about 512 lines. Fields are transmitted at the rate of sixty times per second, to avoid the objectionable flicker effect which occurs at lower rates, even though thirty or fewer images per second are sufficient for motion representation (1).

It is possible to transmit sampled images at reduced bit rates because much of the information in samples taken at the Nyquist rate is redundant. The sucessive samples are not independant, and the video process can be described by a first order Markov model (2). This Markov model fits the measured correlation of the four test images used here, as shown in figure 2 for the the image of newscaster Harry Reasoner, and in table I for all four test images. The image frames usually have the highest correlation between adjacent samples in adjacent lines in a frame (different fields), the next highest correlation between adjacent samples in the same line, and the lowest correlation is between corresponding samples in the closest lines in a field (separated by an alternate field line). This is as expected, from the 4 to 3 width to height aspect ratio of the video frame, and from the fact that the four test frames have low motion or change between fields.

Because higher correlation allows more transmission rate compression, the correlation values indicate that it is most effective to use samples in the adjacent lines of alternate fields, next in effectiveness to use samples in the current line, and least effective to use samples in the closest lines of the same field. The cost of using these samples is the memory required to store them. Using samples in the same line requires a few samples to a line of samples to be stored; using the closest lines in the same field requires several lines of

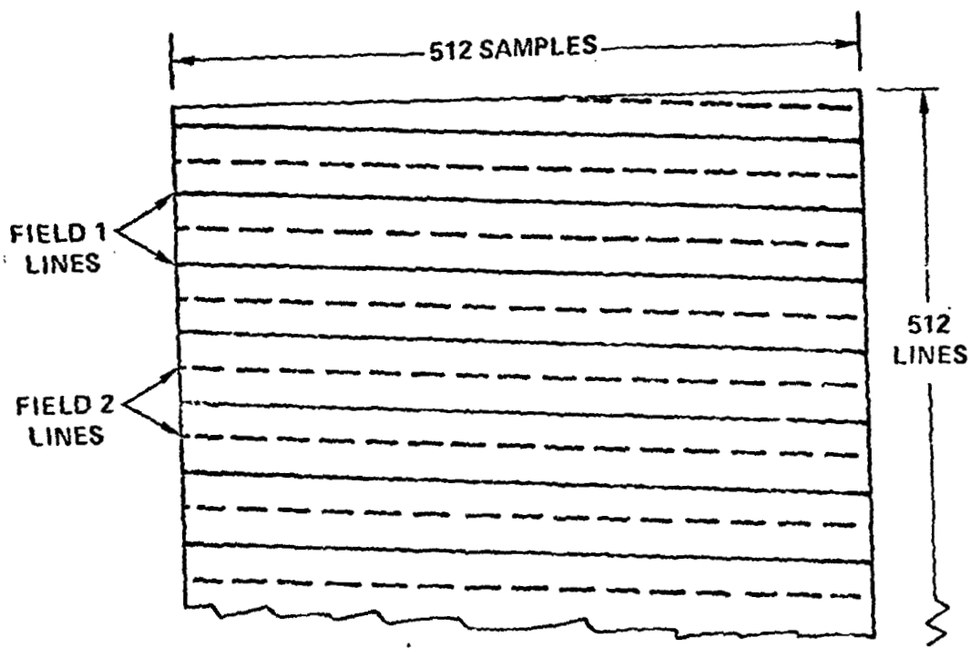
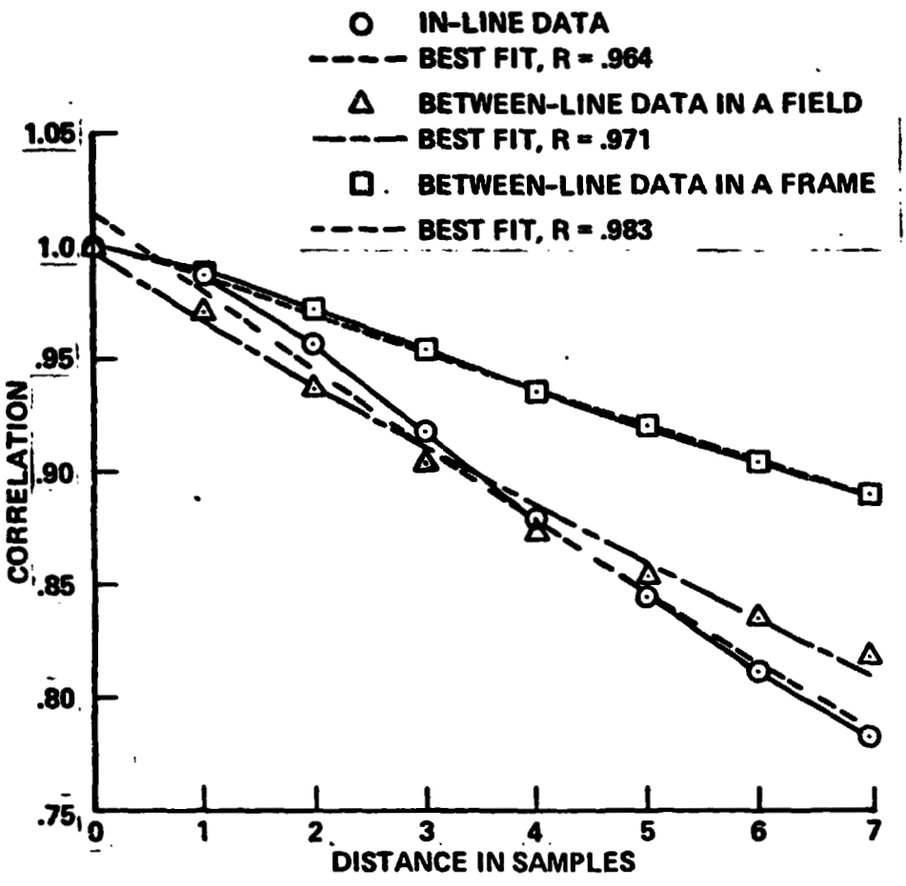


Figure 1: The interlaced video frame, with the first field given by solid lines and the second field given by dashed lines.



JONES

fig 2

Figure 2 . Picture Element Correlation versus Distance for the Reasoner Image.

TABLE I

Range of the correlation parameter R for D equal to 1 through 7, where $R = C^{1/D}$, C is measured correlation and D is sample distance.

Picture	In-line	Between lines in field	Between lines in frame
Reasoner	.966-.987	.967-.973	.984-.989
Two girls	.965-.982	.946-.950	.972-.978
Two men	.964-.977	.933-.946	.968-.972
Band	.847-.882	.807-.877	.888-.916
Assumed in design	.95	.94	.97

memory; and using samples in the adjacent lines of the alternate field requires a full field of memory. Obtaining the lowest possible transmission rate can require much more memory than less efficient systems. This effect is also apparent in conditional replenishment systems, which use the correlation between sucessive frames in time.(3) (4)

TRANSFORM COMPRESSION SYSTEMS

Computer simulations of video image compression systems were undertaken to compare the performance of field and frame compression. All experiments involved single image compression of monochrome television images. Digitized images were obtained by sampling a standard NTSC baseband signal at 8.064 megasamples per second. Each sample was represented by a six bit integer. The visible area of the images has 416 samples per line and 464 lines per frame. The nominal 512 samples and 512 lines includes samples in the horizontal retrace and lines in the vertical interval. The television images were compressed both as fields of 232 lines and as interlaced frames of 464 lines.

The compression experiments used Hadamard transforms of eight by eight subpictures. The coefficients of the sixty-four Hadamard vectors are used to represent the subpictures. The eight by eight Hadamard basis vectors are shown in sequency order in figure 3. Sequency is defined here as the total number of white-black and black-white transitions, in the horizontal or vertical directions. If the Hadamard transform is not normalized, the vector coefficients have a possible range of twelve bits, since each is the result of 64 additions or subtractions of six bit numbers. The vector coefficients were first rounded to the eight most significant bits, and then quantized to an eight bit integer. Transmission rate compression is achieved by using fewer than 64 quantizer levels, and indicating each using a code word shorter than six bits. In the final compressed picture, each sample is represented using a six bit integer, as in the original image.

Figure 4 shows the hardware organization of the independant field transform compressor. The input lines are converted to digital samples, stored in an eight line memory, transformed in eight sample by eight line blocks, and quantized.

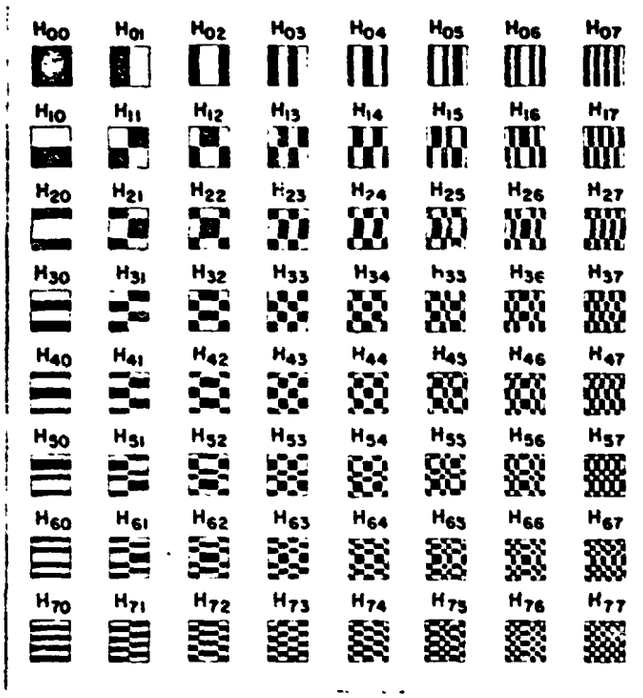


Figure 3: The sequency ordered 8 by 8 Hadamard vectors.

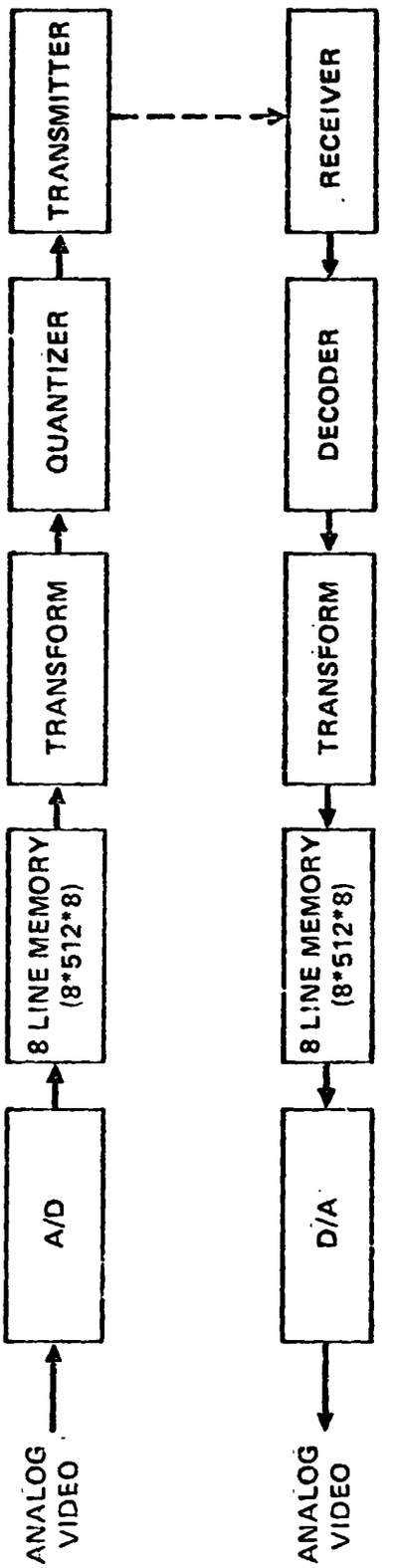


Figure 4: Field transform compression.

The quantized bit stream is transmitted, and the inverse process is used to generate analog video. The field compressor uses the correlation between the samples in a line and between the lines in a field. Some correlations are not used, since each eight by eight subpicture is processed independantly.

Figure 5 shows the interlaced frame transform compressor, which performs the same functions as the field processor. It differs because eight by eight subpictures are taken from an interlaced frame, rather than one field. The eight by eight subpictures have one-half the height of field subpictures. In order to interlace a frame, the first field is held in memory until the second field is being generated. The fields are then interlaced and the subpictures are transformed. Information on the two fields is partly transmitted and partly stored during the second field time, and the stored information is transmitted during the next field time. The receiver output display is not synchronized to the data transmission, as it was in the field compressor. To provide the correct display, the receiver requires a compressed memory to hold the frame in the transmitted form. This memory is decoded twice, to provide the two fields. The memory required is one field at eight bits and one-half of a compressed frame (assumed to require 1 bit per sample) at the encoder, and one compressed frame at the decoder (assumed to require 2 bits per sample), or the equivalent of one frame at 7 bits (i.e., 7 bit frames). This is the cost of using the correlation between adjacent lines in a frame, rather than the correlation between the closest lines in a field.

SYSTEM SIMULATION RESULTS

Figure 6 shows the mean-square error results obtained (in units of the least significant of the six original bits) when the Harry Reasoner test image was compressed using theoretical compression designs. The different compression designs consist of the bit assignments and quantizers for each of the 64 Hadamard vector coefficients. The theoretical designs assumed the first order Markov correlation model (with the "assumed in design" values of table I), an exponential distribution for the transform vector coefficients, and the mean-square error measure. At the same rates, field compression produces larger error than frame compression, or, equivalently, field compression requires more rate for a given error. However, there are two cases in the field data, and one case in the frame data, where 1/2 or 1 bit per sample increases in the transmission rate produce

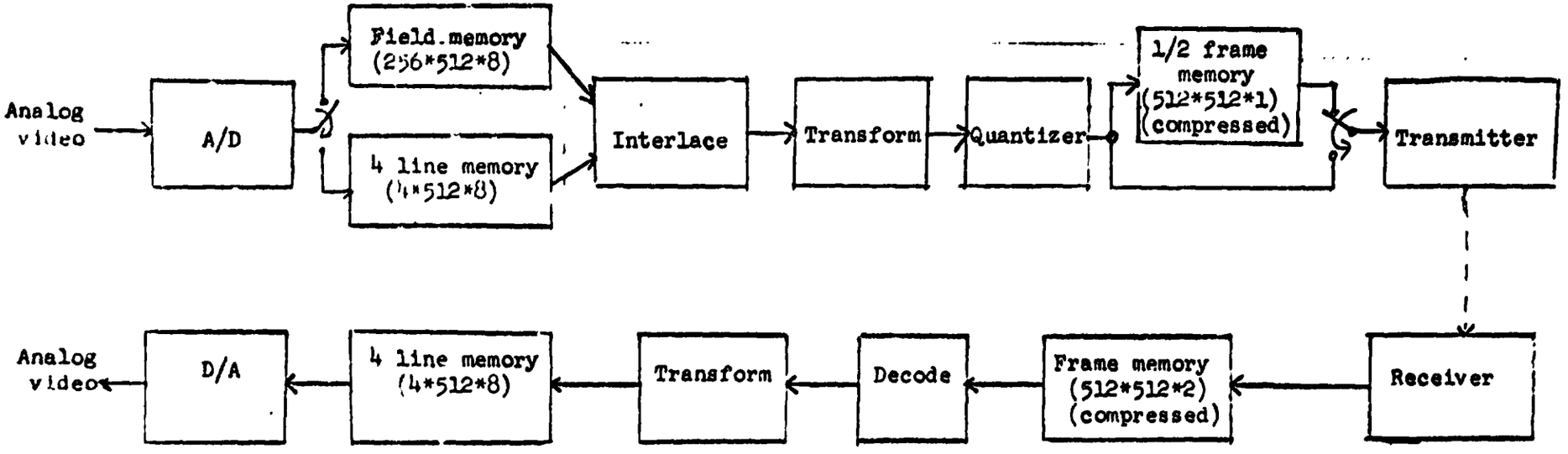
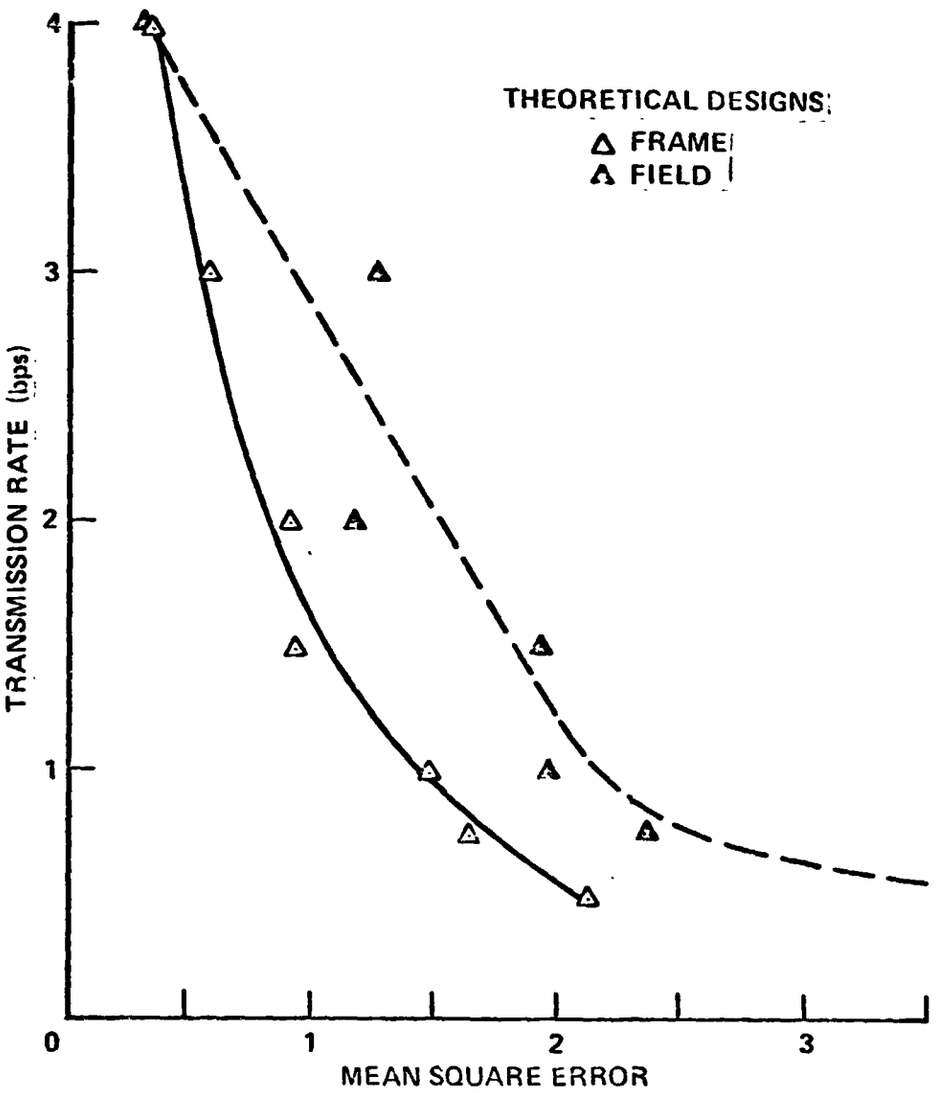


Figure 5: Frame transform compression (7 bit frames of memory).

Figure 6: Rate versus mean-square error for the Harry Reasoner image compressed using theoretical compression designs.



little or no reduction in error. The theoretical designs obviously do not make the best possible use of the transmission rate.

Figure 7 shows the mean-square error obtained when the Harry Reasoner test image was compressed using experimental compression designs. The curves are smooth, and added rate always reduces error. The experimental designs give much lower error than the theoretical designs. The field compression curve for the experimental designs (figure 7) is nearly identical to the frame curve for the theoretical designs (figure 6), from 4 bits per sample down to one bit per sample. The experimental designs used are similar to designs obtained by trial and error, but were generated using a formalized procedure based on the requirement of good representationfor both the edges and the low detail areas in video images. For a full discussion of the theoretical and experimental designs used, see reference 5.

Figure 7 shows that, over most of the range of transmission rates, field compression requires a transmission rate about fifty percent greater than frame compression, for the same mean-square error. At the highest rate shown, 4 bits per sample, the mean-square error is caused by rounding all the transform vectors to eight bits, and all methods give about the same error.

Figure 8 shows the mean-square error obtained using the experimental compression designs on all four test images, in both frame and field compression. Because of the wide range in the detail and correlation of the test images, the mean-square error at each transmission rate ranges over an order of magnitude, and the the mean-square error is plotted on a log scale. Even though the test images differ greatly, the parallel curves of figure 8 show that the increased rate required by field compression is nearly constant, at about 1/2 bit per sample, for these images in the range of transmission rates between 1/2 and 2 bits per sample. It seems that the frame or field compression trade-off can be summarized as 7 bit frames of memory for 1/2 bit per sample in transmission rate.

The subjective impressions of the compressed images agree in quality ranking with the mean-square error results. Figure 9 shows the original image of Harry Reasoner. Figure 10 shows this image compressed using 1 bit per sample in the

Figure 7: Rate versus mean-square error for the Harry Reasoner image compressed using experimental compression designs.

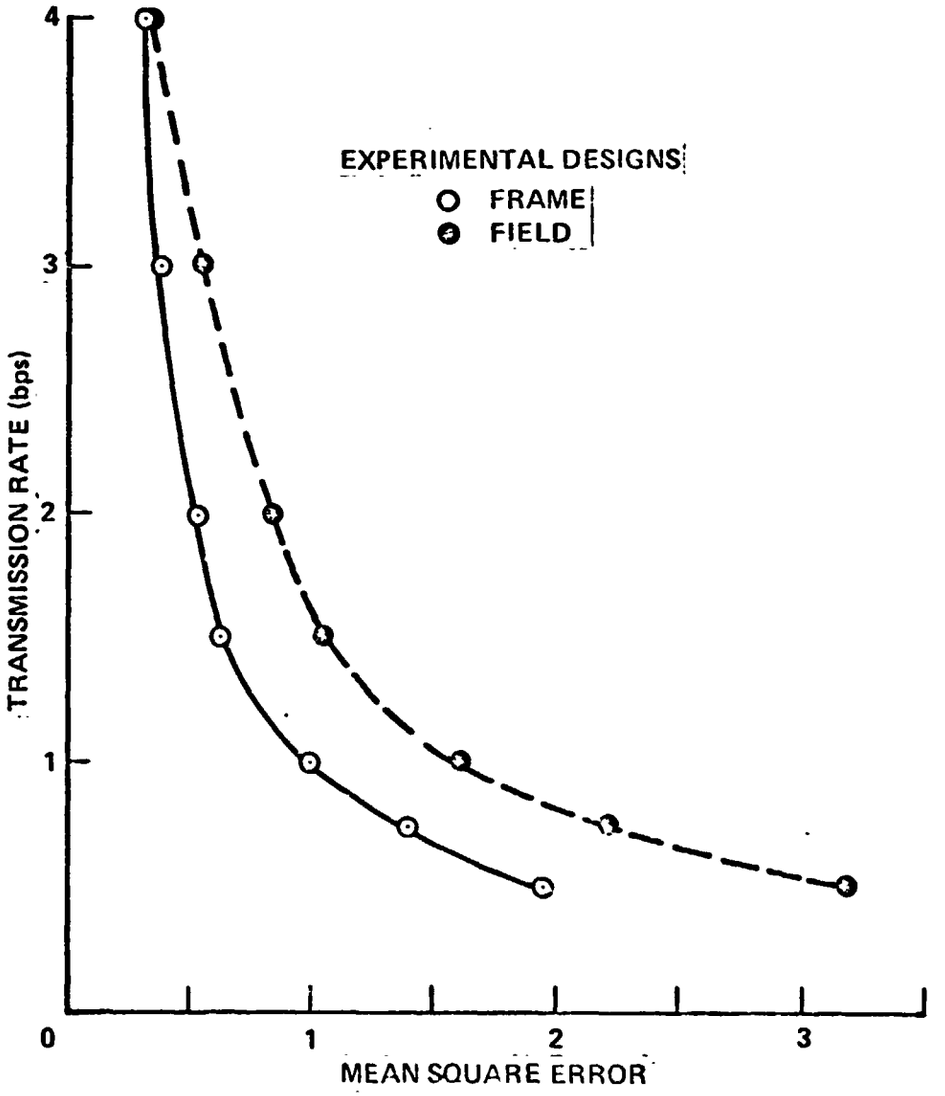
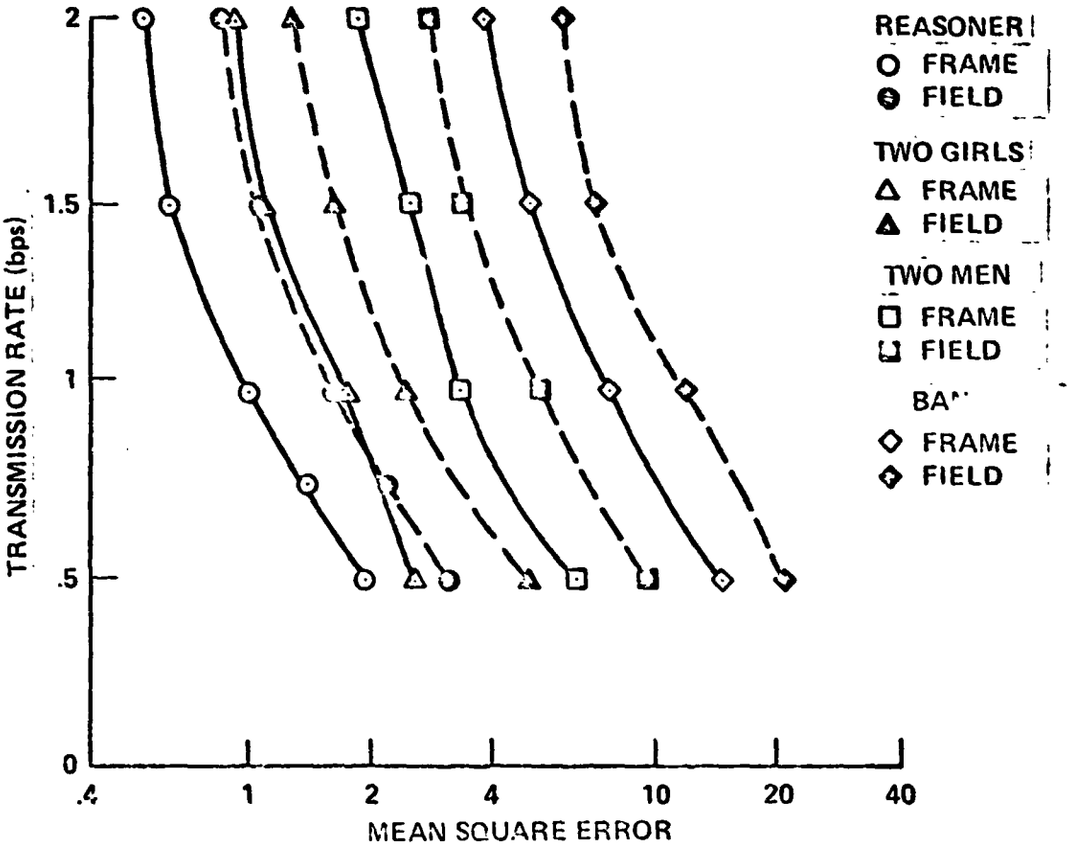
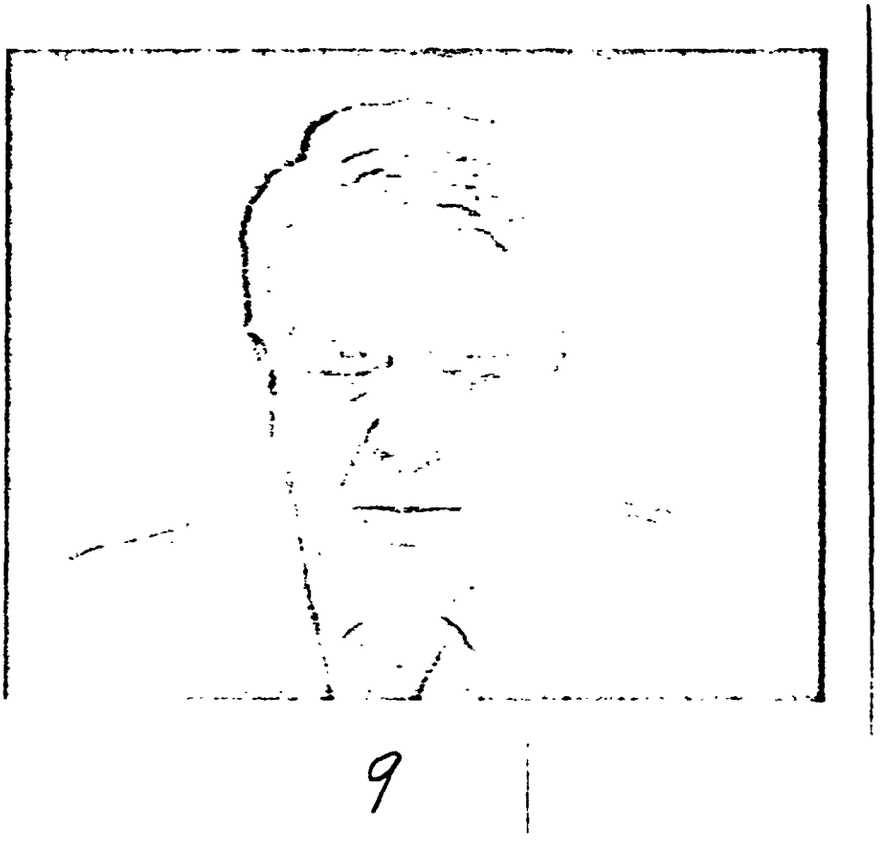


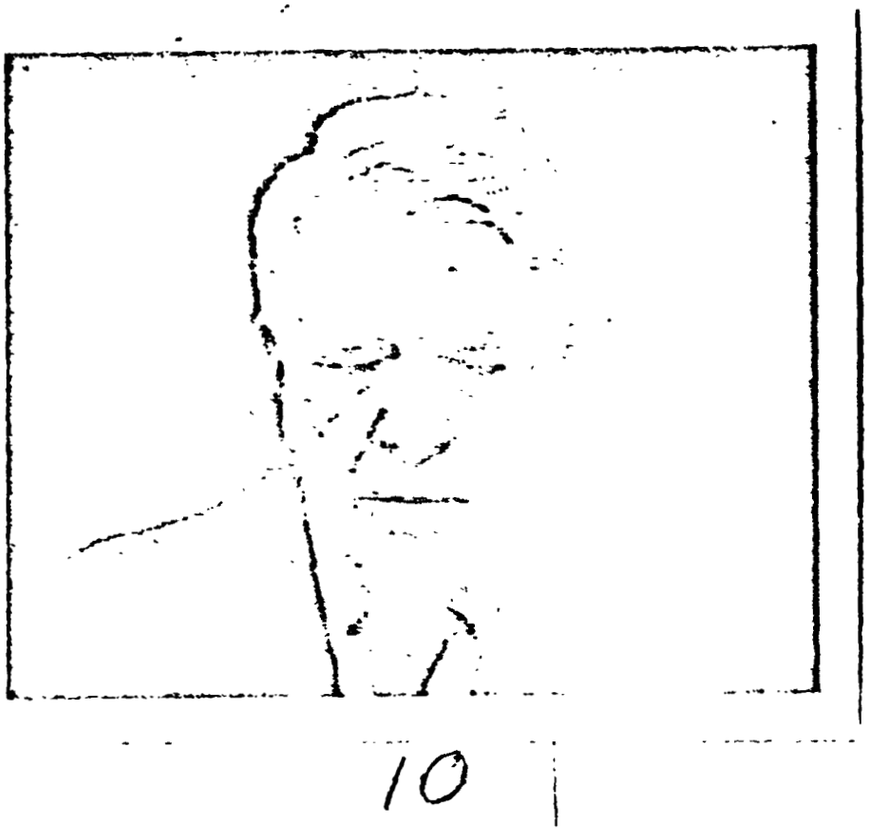
Figure 8: Rate versus mean-square error for the four test images compressed using experimental compression designs.





9

Figure 9: The original Reasoner image.



10

Figure 10: The Reasoner image processed as an interlaced frame at 1 bpp, using the experimental design.

frame, and figure 11 shows it compressed using 2 bits per sample in the field. The compressed images exhibit edge degradation, especially at the shoulders, lips, collar, and tie. The field image at 2 bits per sample has somewhat higher quality than the frame image at 1 bit per sample, as indicated by the error values of figures 7 and 8. (An original of the band image is shown in reference 6.)

TIME EFFECTS

The above comparison of frame and field video compression considered only the quality of the individual images, and ignored the effects of motion and the time sequence of images. The two fields in a frame are generated one-sixtieth of a second apart in time, and motion tends to make the correlation between adjacent lines in a frame lower than the correlation between the closest lines in the same field. A fifth test image, of a blurred hand moving rapidly over a writing pad, was compressed in the same way as the four other test images. Because of the motion, the mean-square error was lower for field compression. (An original of the pad image is shown in reference 6.) Transform compression, especially at lower transmission rates, tends to average adjacent samples and lines. Two fields processed as a frame become similar, and high motion areas where the original fields differ become blurred. In frame compression of high motion scenes at low transmission rates, the motion update rate is the frame rate, thirty per second, rather than the field rate, sixty per second. Because the frame rate is adequate for representing motion, this is not an impairment.

These observations suggest a third approach to video compression. Since frame processing tends to average the two fields at lower transmission rates (which would reduce the vertical resolution and the motion update rate to one-half their original values), frame compression is more similar to field repeat compression than to independant two field processing. In field repeat compression, only one-half the fields are transformed and transmitted, and each transmitted field is displayed twice at the decoder. Figure 12 shows the block diagram of a field repeat compression system. As the field to be transmitted is generated,

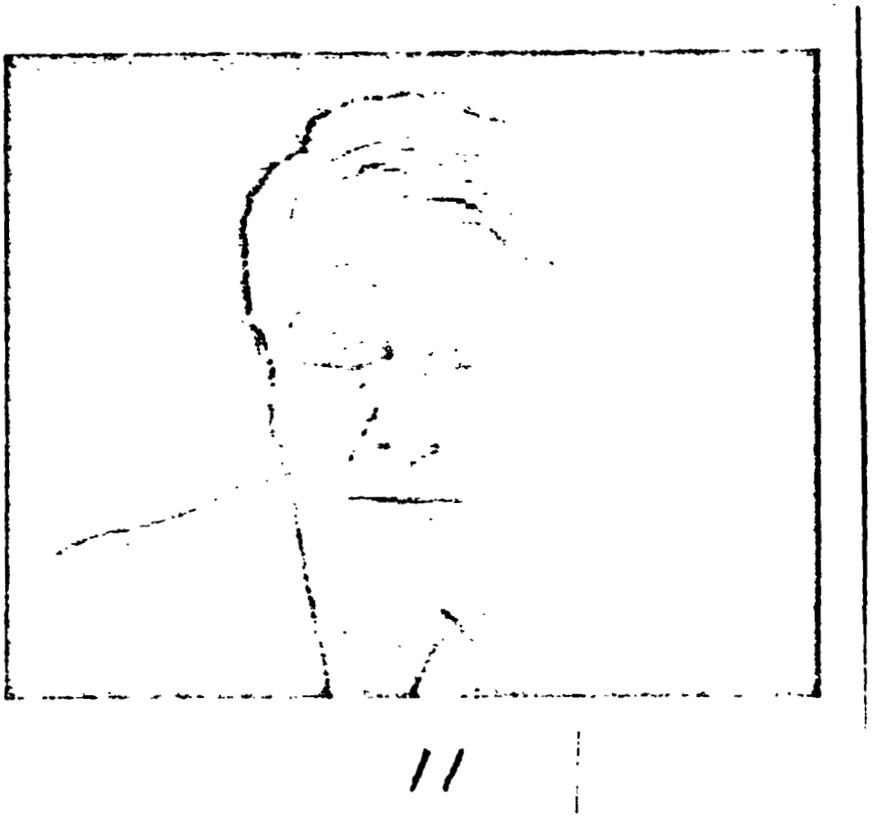


Figure 11: The Reasoner image processed as two fields at 2 bpp, using the experimental design.

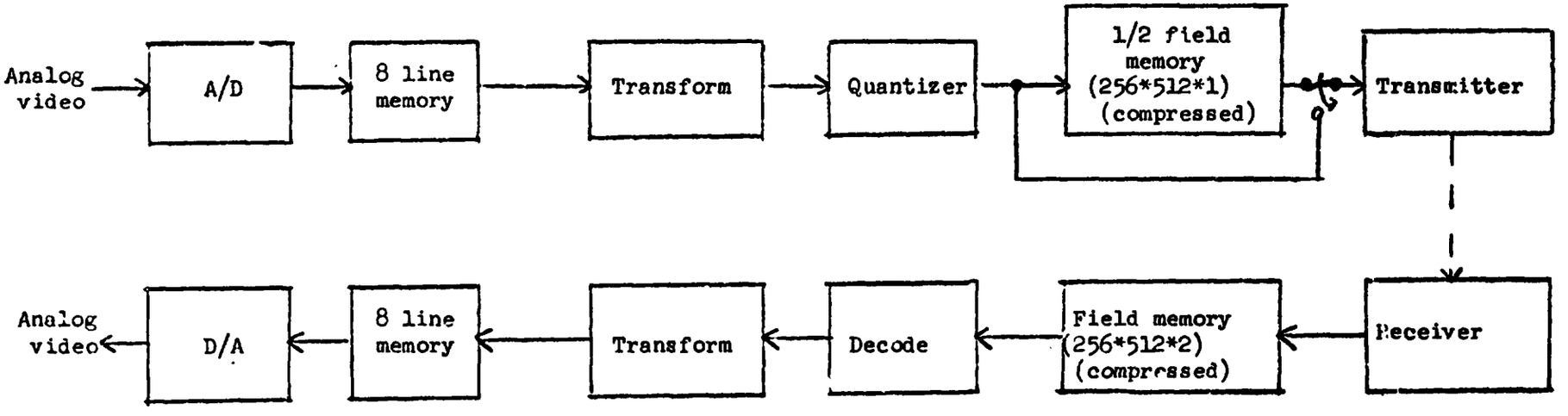
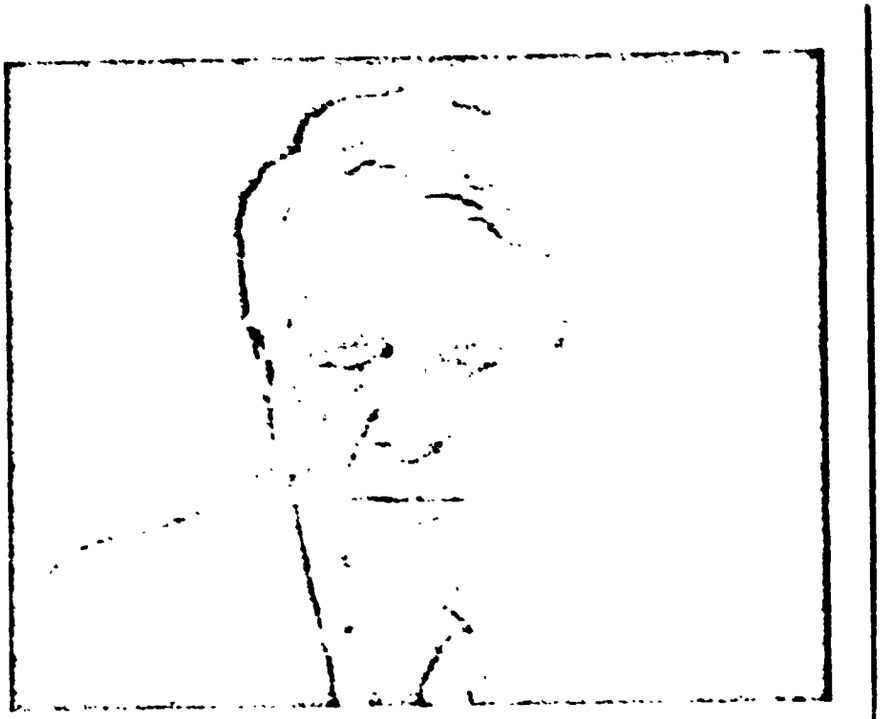


Figure 12. Field repeat compression (1 1/2 bit frames of memory)

half the current information is transmitted and half is stored. The full compressed field is retained in the decoder, for repeated display. The total memory requirement for field repeat compression is 1 1/2 bit frames, rather than 7 bit frames for frame compression. A real time hardware system using the Hadamard transform and field repeat has been previously described. (6)

Since a single field has only one-half of the samples in a frame, the same overall transmission rate is obtained when the number of bits per transmitted sample is doubled for field repeat. The overall rate is the number of bits per image multiplied by the number of images per second. Field repeat compression transmitts only one-half of the field images used in frame or field compression, as discussed above. The previous mean-square error results also indicate the performance of field repeat, since the same error in each transmitted field is obtained if one or two independant fields are transmitted. Field repeat compression at 2 bits per sample has the same error as field compression at 2 bits per sample, but the overall transmission rate for field repeat corresponds to that for frame or field compression at 1 bit per sample. Field repeat compression can be compared to frame or field compression in figures 6, 7, and 8 by moving each point on the field compression curve to a point having the same mean-square error and one-half the transmission rate. This shows that the error is slightly lower for field repeat compression than for frame compression, and much lower than for field compression.

Figure 13 shows a field repeat image of the first field of figure 11. This field repeat image requires the same overall transmission rate as the frame processed image in figure 10, having one field at 2 bits per sample rather than one frame at 1 bit per sample, and the subjective quality is similar. The field repeat image has lower quality than the field compressed image, but that image has two independant fields at 2 bits per sample, and requires twice the overall transmission rate. In field repeat, vertical resolution is noticeably reduced in detailed areas, and quantization noise and contouring are more apparent in background areas. It should be reemphasized that field repeat compression is appropriate at the lower transmission rates, where it is not possible to provide the full potential resolution of uncompressed video.



13

Figure 13: The Reasoner image using field repeat. The first field is compressed using 2 bpp as in fighre 11, and the second field is supplied by repeating the first field.

CONCLUSION

Experimental simulations of interlaced frame and independant field compression systems indicate that frame compression can achieve a transmission rate about one-half bit per sample lower than field processing at a given image quality, with the added requirement of 7 bit frames of memory. Frame transform compression can be used at lower transmission rates than field compression, but replaces the two fields in the frame with two similar combinations of the original fields. If it is decided to use only one field in field repeat compression, performance similar to frame compression at low transmission rates can be obtained using only 1 1/2 bit frames of memory. A conditional replenishment compressor, which uses the correlation between sucessive frames, can be implemented using 7 bit frames of memory. (4)

ACKNOWLEDGEMENT

The author is grateful for valuable discussions with Larry B. Hofman and Scott C. Knauer of the NASA Ames Research Center, and for the use of the NASA Ames Image Processing Facility.

REFERENCES

1. R.C. Brainard, F.W. Mounts, and B. Prasada, "Low-Resolution TV: Subjective Effects of Frame Repetition and Picture Replenishment," Bell System Technical Journal, Jan. 1967, pp. 261-271.

2. L.E. Franks ,"A Model for the Random Video Process," Bell System Technical Journal, April 1966, pp. 609-630.

3. B.G. Haskell,"Interframe Coding of Monochrome Television - A Review," SPIE vol. 87, Advances in Image Transmission Techniques (1976), pp. 212-221.

4. H.W. Jones,"A Conditional Replenishment Hadamard Video Compressor," SPIE vol. 119, Applications of Digital Image Processing (IOCC 1977), pp. 91-98.

5. H.W. Jones, "A Comparison of Theoretical and Experimental Video Compression Designs," to be published in the IEEE Trans. on Electromagnetic Compatibility.

6. H.W. Jones, "A Real-Time Adaptive Hadamard Transform Video Compressor," SPIE vol. 87, Advances in Image Transmission Techniques (1976), pp. 2-9.

APPENDIX B

THE K-L, DCT, AND RELATED TRANSFORMS

OBTAINED VIA THE HADAMARD TRANSFORM

I. INTRODUCTION

It is well known that the Karhunen-Loeve or eigenvector transformation provides the maximum possible data compression, and also that the discrete cosine transform is a close approximation to the Karhunen-Loeve transformation, for highly correlated data fitting the first-order Markov model. For less correlated data, the discrete cosine transform provides data compression nearly equal to that of the Karhunen-Loeve transform, even though the transforms differ. The Hadamard transform provides most of the potential data compression, but it always provides less data compression than the discrete cosine transform for the first-order Markov model. There is a general class of transforms, which are computed via the Hadamard transform, that can be designed to approach the performance of the Karhunen-Loeve transform, and also to meet various restrictions which simplify hardware implementation.

II. THE KARHUNEN-LOEVE TRANSFORM

In the proof that the Karhunen-Loeve transform (KLT) is the optimum orthonormal transform with respect to the mean-square distortion measure, it is assumed that distortion is introduced by neglecting to transmit some of the lower energy transform coefficients.[1] The KLT is optimum because the distortion, which is the total energy of the neglected coefficients, is minimum for any number of neglected coefficients. If the transmitted coefficients are described with finite accuracy, the optimum coefficient bit assignment is well known.[2]

$$b_i = \frac{1}{2} \log_2 (\sigma_i^2/d) \qquad \sigma_i^2 > d$$
$$= 0 \qquad \sigma_i^2 \leq d \qquad (1)$$

σ_i^2 is the energy of the ith transform vector coefficient, and b_i is the number of bits used to transmit the ith coefficient. b_i is zero when the coefficient energy is less than the distortion, d, allocated for each vector. If M of the N vectors are transmitted, the distortion is

$$D = M d + \sum_{i=M+1}^{N} \sigma_i^2 \qquad (2)$$

The total bit rate for M transmitted vectors is

$$B = \sum_{i=1}^{M} b_i = \sum_{i=1}^{M} \frac{1}{2} \log_2 (\sigma_i^2/d)$$
$$= \frac{1}{2} \sum_{i=1}^{M} \log_2 \sigma_i^2 - \frac{M}{2} \log_2 d$$
$$= \frac{1}{2} \log_2 \prod_{i=1}^{M} \sigma_i^2 - \frac{M}{2} \log_2 d \qquad (3)$$

Under the constraint that the energy of the transmitted vectors is fixed, or that the distortion is fixed, we can show that the total rate, B, is minimum for the KLT. Suppose that the transform vectors are ranked in order of decreasing energy, and that a suboptimal transform is used. For some i less than j, the energy, e, is transferred from σ_i^2 to σ_j^2. From equation 3, B is reduced by the suboptimal transform only if

$$(\sigma_i^2 - e)\,(\sigma_j^2 + e) < \sigma_i^2\,\sigma_j^2$$

$$\sigma_i^2\, e - \sigma_j^2\, e - e^2 < 0$$

$$\sigma_i^2 < \sigma_j^2 + e$$

Since this implies that the suboptimal transform has more energy compaction than the KLT, which is impossible, the rate for fixed distortion is minimized by the KLT.

If $\tilde{\tilde{\Sigma}}_x$ is the correlation matrix of the data sample vector, and $\tilde{K}$ is the matrix of the KLT, the correlation matrix of the KLT vectors is

$$\tilde{\tilde{\Sigma}}_k = \tilde{K}\,\tilde{\tilde{\Sigma}}_x\,\tilde{K}^{-1}$$

$\tilde{\tilde{\Sigma}}_k$ is the diagonal matrix $(\sigma_1^2, \sigma_2^2, \ldots, \sigma_N^2)$, where the vector energies σ_i^2 are the eigenvalues of $\tilde{\tilde{\Sigma}}_x$.

III. THE KARHUNEN-LOEVE TRANSFORM OBTAINED VIA THE HADAMARD TRANSFORM

The eigenvectors, the rows of $\tilde{K}$, are even and odd vectors which can be obtained independantly from the odd and even Hadamard vectors. The general implementation of the KLT requires N^2 multiplications. These can be replaced by a Hadamard transform followed by $N^2/2$ multiplications.

The elements of $\tilde{\Sigma}_x$ are the sample correlations, $c(x_i,x_j)$, where i and j, $1 \leq i,j \leq N$, indicate the sample locations. Suppose the correlation is stationary, $c(x_i,x_j) = c(|i-j|)$. $\tilde{\Sigma}_x$ has the form [3, p109].

$$\tilde{\Sigma}_x = \begin{vmatrix} c(0) & c(1) & c(2) & \dots & c(N) \\ c(1) & c(0) & c(1) & \dots & c(N-1) \\ \dots & & & & \\ c(N) & c(N-1) & c(N-2) & \dots & c(0) \end{vmatrix} = \begin{vmatrix} \underline{cr1} \\ \underline{cr2} \\ \vdots \\ \underline{crN} \end{vmatrix}$$

Row cr1 is the transpose of row crN, and row cri is the transpose of row cr(N+1-i).

The Walsh-Hadamard transform (WHT) is defined for N a power of 2. [1] The WHT is orthonormal. The rows and columns of the WHT matrix, $\tilde{H}$, are the transform vectors, and have sample weighting values of +1 and -1. The WHT vectors are even and odd. $\tilde{H}^T$ is written as columns of orthonormal transform vectors

$$\tilde{H}^T = \begin{vmatrix} hc1 & hc2 & \dots & hcN \end{vmatrix}$$

We assume that $\tilde{H}$ is in bit reverse order [1] so that the first N/2 columns are even vectors, and the second N/2 are odd vectors.

We compute the WHT correlation matrix, to show that the even and odd WHT vectors are uncorrelated.

$$\tilde{\tilde{\Sigma}}_H = \tilde{H}\,\tilde{\Sigma}_x\,\tilde{H}^{-1}$$
$$= \hat{H}\,\tilde{Y}$$

$$\tilde{Y} = \tilde{\tilde{\Sigma}}_x \tilde{H}^T = \left|\begin{array}{c}\text{cr1}\\ \hline \text{cr2}\\ \hline \vdots \\ \hline \text{crN}\end{array}\right| \quad \Big|\,\text{hc1}\,\Big|\,\text{hc2}\,\Big|\ldots\Big|\,\text{hcN}\,\Big|$$

$$\tilde{Y} = \begin{vmatrix} y_{11} & y_{21} & \cdots & y_{N1} \\ y_{12} & y_{22} & \cdots & y_{N2} \\ \cdots & & & \\ y_{1N} & y_{2N} & \cdots & y_{NN} \end{vmatrix}$$

Because cri is the transpose of cr(N+1-i), y_{ji} = cri hcj = cr(N+1-i) hcj = $y_{j,N+1-i}$ whenever hcj is even, and y_{ji} = cri hcj = - cr(N+1-i) hcj = - $y_{j,N+1-i}$ whenever hcj is odd. Writing $\tilde{Y}$ in terms of columns

$$\tilde{Y} = \Big|\,\text{yc1}\,\Big|\,\text{yc2}\,\Big|\ldots\Big|\,\text{ycN}\,\Big|$$

yci is even if hci is even; yci is odd if hci is odd. Since the rows of $\hat{H}$ are the same as the columns of $\tilde{H}$, and since even and odd vectors are orthogonal, it follows that half the elements of $\tilde{\tilde{\Sigma}}_H$ are zero.

$$\tilde{\tilde{\Sigma}}_H = \tilde{H}\,\tilde{\tilde{\Sigma}}_x\,\tilde{H} = \tilde{H}\,\tilde{Y} = \left|\begin{array}{c:c} \tilde{\tilde{\Sigma}}_{H1} & \tilde{0} \\ \hdashline \tilde{0} & \tilde{\tilde{\Sigma}}_{H2} \end{array}\right|$$

The even WHT vectors are uncorrelated with the odd vectors, and the completely uncorrelated KLT vectors can be obtained by operating independantly on the even and WHT vector sets. #

$$\tilde{\Sigma}_K = \tilde{K}\tilde{\Sigma}_x\tilde{K}^{-1} = \tilde{A}\tilde{H}\tilde{\Sigma}_x\tilde{H}^{-1}\tilde{A}^{-1}$$

$$\tilde{A} = \left|\begin{array}{c|c} \tilde{A}_1 & 0 \\ \hline 0 & \tilde{A}_2 \end{array}\right| \qquad (4)$$

Since a linear combination of even/odd vectors is even/odd, the KLT vectors are even and odd. The KLT can be obtained, following the WHT, by $2\,(N/2)^2 = N^2/2$ multiplications. This matrix factorization is the basis of implementation of approximations to the KLT described below.

The separation of vectors into even and odd groups occurs in the first operation of the bit reverse WHT. [1, p106]

IV. THE DISCRETE COSINE TRANSFORM OBTAINED VIA THE HADAMARD TRANSFORM

The discrete cosine transform (DCT) also has even and odd vectors[1,4], and Hein and Ahmed have shown how the DCT vectors can be obtained by a sparse matrix multiplication on the WHT vectors.[5] Since the DCT, unlike the general KLT, has a constant vector and a shifted square wave vector in common with the WHT, the number of matrix multiplications is fewer than $N^2/2$. The $\tilde{A}$ matrix (equation 4) which generates the DCT vectors for N =8 from the WHT vectors is given by Hein and Ahmed, and is reproduced here as figure 1. While this implementation of the DCT requires more operations for large N than the most efficient DCT implementations[6], it is very satisfactory for smaller N.

If a transform has even and odd vectors and has a constant vector, as is typical, it can be obtained via the WHT in the same way as the DCT. The slant transform is an example.[7,1] A hardware implementation of the DCT via the WHT is being constructed at NASA-Ames Research Center, using N = 8 and the matrix of figure 1. Since this implementation contains the matrix multiplication factors in inexpensive read-only memories, it will be possible to consider the real-time quantization design and evaluation of a large class of transforms. Transforms with suboptimum performance are acceptable only if they can be implemented with reduced complexity. Transform performance can be determined theoretically from the vector energy compaction, while the implementation complexity can be estimated from the number and type of operations added after the WHT.

$$\tilde{A} = \begin{bmatrix} 1.0 & & & & & & & \\ & 1.0 & & & & & & \\ & & 0.923 & 0.383 & & & & \\ & & -0.383 & 0.923 & & & & \\ & & & & 0.907 & -0.075 & 0.375 & 0.180 \\ & & & & 0.214 & 0.768 & -0.513 & 0.318 \\ & & & & -0.318 & 0.513 & 0.768 & 0.214 \\ & & & & -0.180 & -0.375 & -0.075 & 0.907 \end{bmatrix}$$

Figure 1: The $\tilde{A}$ matrix used to obtain the DCT from the WHT.

V. COMPARISON OF TRANSFORMS USING THE FIRST ORDER MARKOV CORRELATION MODEL

It is generally accepted that the sample-to-sample correlation of an image line scan is approximated by the first-order Markov model.[8]

$$c(x_i,x_j) = c(\,|i-j|\,) = r^{|i-j|} \qquad (5)$$

r is the corelation of adjacent samples, and varies from 0.99 for low detail images to 0.80 for high detail images, with an average of about 0.95.[9] The correlation matrix, Σ_x, was generated using the first-order Markov model, for various r, and the corresponding KLT's and vector energies were numerically computed.# In addition, the matrix Σ_x was used to compute the transform vector energies and correlations for the WHT, DCT, and other transforms, using equation 4 above.

As is well known, the KLT vectors for r = 0.95 are very similar to the DCT vectors, and have nearly identical vector energies.[14] The most apparent difference between the DCT and the KLT is that the KLT vector corresponding to the constant DCT vector is not exactly constant, but weights the central samples in a fixed transform block more than samples near the edge of the block. As r approaches 1.00, this KLT vector approaches the constant vector, and all the KLT vectors approach the corresponding DCT vectors. As r becomes less than 1.00, the higher weighting of the central samples increases. The weights of the first and fourth samples, for N =8, are given in table I. As the sample correlation decreases, the central samples provide a better estimate of the average sample value than do the extreme samples. The vector energies of the KLT and the DCT are nearly identical for r greater than 0.90, and differ only slightly for r greater than 0.50. The KLT and DCT vector energies for N = 8 and r =0.50 are plotted in figure 2. The energy compaction at r = 0.5 is much less than at the typical r = 0.95.

The analytic solution is known.[10]

Table I. The weights of the first and fourth samples in the KLT, for $N = 8$, and various correlations

Correlation r	Weight of the first sample	Weight of the fourth sample
0.999	0.353	0.354
0.99	0.350	0.356
0.95	0.338	0.364
0.9	0.324	0.374
0.8	0.296	0.392
0.7	0.272	0.407
0.6	0.250	0.420
0.5	0.230	0.430

ORIGINAL PAGE IS
OF POOR QUALITY

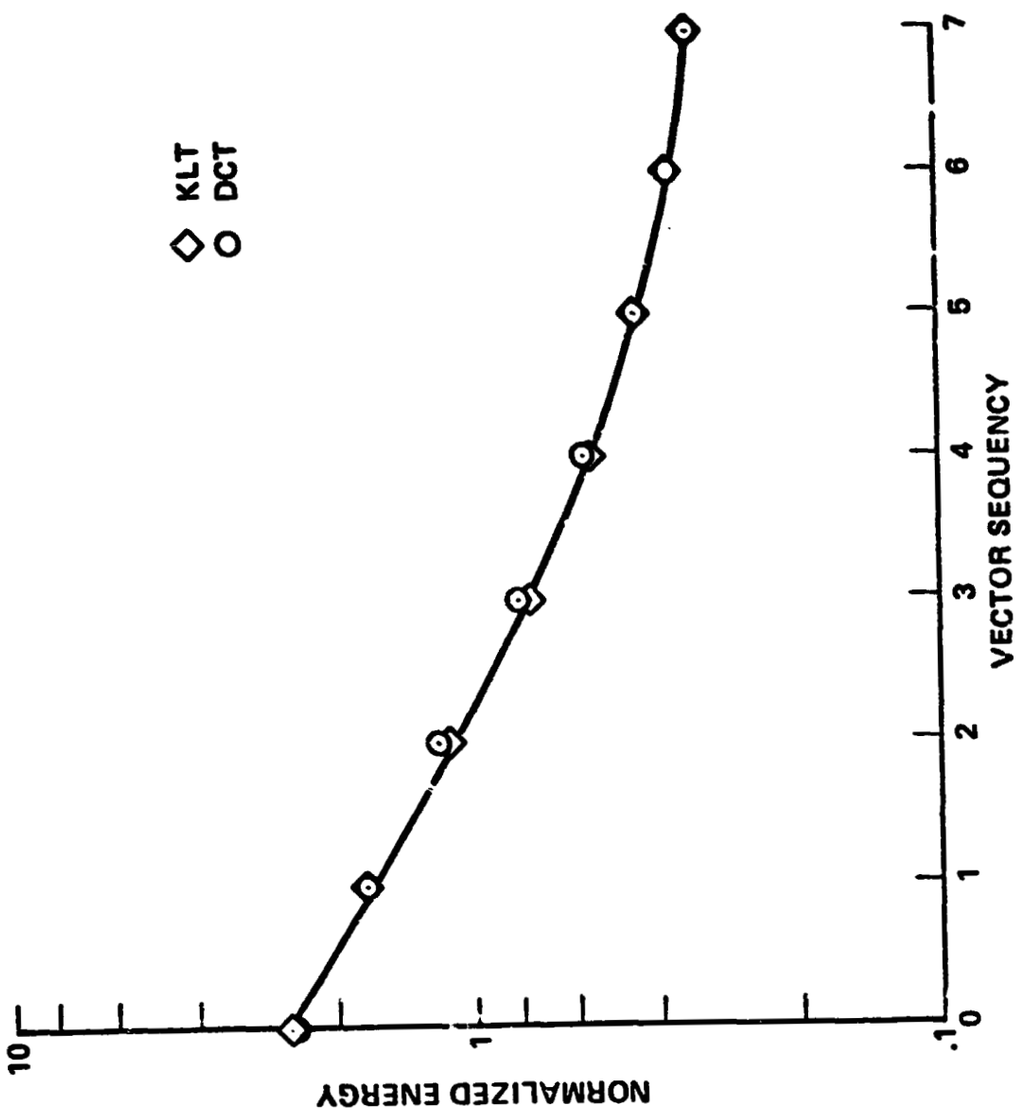


Figure 2. KLT and DCT vector energies for N=8 and r=0.5

The rate-distortion performance of a transform depends on the transform energy compaction, as shown in equations 2 and 3. For small distortion, d is less than σ_i^2 for all i, and all N transform vectors are quantized and transmitted. The number of bits required is

$$B = \frac{1}{2} \sum_{i=1}^{N} \log_2 \sigma_i^2 - \frac{N}{2} \log_2 d \qquad (3)$$

The first term of B can be used as a figure of merit for a transform.

$$F = \frac{1}{2} \sum_{i=1}^{N} \log_2 \sigma_i^2$$

F is a negative number; the larger its magnitude, the greater the rate reduction achieved by the transform. Table II gives the rate reduction factor, F, for the KLT, DCT, WHT, and two similar transforms that will be described below. At correlation r = 0.95, the KLT gains 0.014 bits more than the DCT, and 1.183 bits more than the WHT. The WHT achieves most of the available data compression, and the DCT achieves nearly all. As this rate reduction is obtained for all N vectors, the increased compression of the DCT over the WHT, for r = 0.95, is 1.169/8, or 0.15 bits per sample.

Table II. The figure of merit, $F = \Sigma \log_2 \sigma_i$, for different transforms at N = 8 and various correlations.

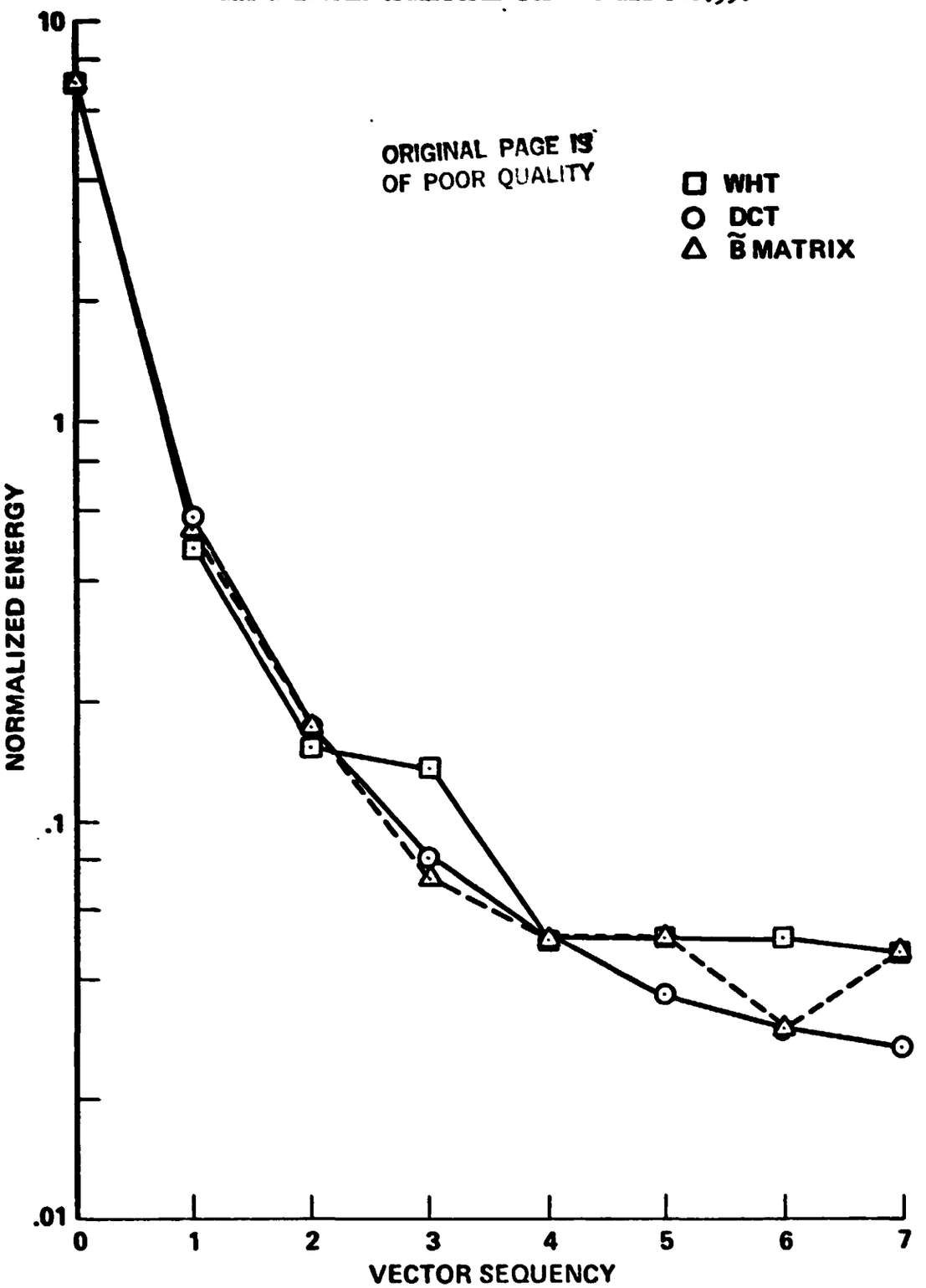
Correlation, r	Transform				
	KLT	DCT	WHT	$\tilde{B}$ matrix	$\tilde{C}$ matrix
0.99	-19.817	-19.775	-18.489	-19.205	-19.597
0.95	-11.743	-11.729	-10.560	-11.206	-11.558
0.90	-8.379	-8.341	-7.311	-7.875	-8.180
0.80	-5.162	-5.092	-4.317	-4.731	-4.954
0.70	-3.402	-3.328	-2.765	-3.056	-3.214
0.50	-1.453	-1.396	-1.136	-1.261	-1.333
0.00	0.00				

VI. OTHER TRANSFORMS OBTAINED VIA THE HADAMARD TRANSFORM

If a transform with good performance and simpler implementation than the DCT is required, approximations to the DCT obtained via the WHT can be considered. The matrix multiplication of the WHT vectors which produces the DCT for N = 8, first given by Hein and Ahmed, is shown in figure 1 above. The sequency of a transform vector is defined as the number of sign changes in the vector. The vector sequencies of the vectors corresponding to the matrix of figure 1 are in bit reverse order, as indicated (0,4,2,6,1,5,3,7). The energy compaction of the WHT and DCT for r = 0.95 and N = 8 is shown in figure 3. In the conversion from WHT to DCT, the two by two matrix operation on vectors 2 and 6 transfers energy from 6 to 2. The four by four matrix operation on the vectors of sequency 1, 5, 3, and 7 reduces the energy of 3, 5, and 7 and increases the energy of 1. These operations remove most of the residual correlation of the WHT vectors. The matrix multiplication requires twenty multiplications, by ten different factors (fifteen factors including sign differences).

We first consider a simplified operation on the 2 and 6 and the 1 and 3 sequency vectors. This operation consists of multiplying the WHT vectors by matrix B, given in figure 4. This further transform is designed to reduce correlation and to generate new transform vectors in a way somewhat similar to the A matrix multiplication which produces the DCT. There are two identical two by two operations, and a total of eight multiplications by two different factors (three including sign). The energy compaction of the B matrix transform is shown in figure 3, with the energies of the WHT and DCT. As the B matrix transform vectors of sequency 0, 4, 5, and 7 are identical to the WHT vectors, they have identical energy. The B matrix transform vectors of sequencies 0, 1, 2, 3, 4, and 6 are identical to the corresponding DCT vectors (0, 4) or very similar. For example, the B matrix vector of sequency 1 is a slanted vector of step width 2 and step size 2 (3,3,1,1,-1,-1,-3,-3). The performance of the B matrix transform, in terms of the figure of merit, is given in table II above. The B matrix transform has something more than one-half of the gain of the DCT over the WHT, with something less than one-half the multiplications, and less than one-fourth the hardware if the two by two transformer is used twice.

Figure 3. The energy compaction of the DCT, WHT, and $\tilde{B}$ matrix transforms for N=8 and r=0.95.



$$\hat{B} = \begin{bmatrix} 1.00 & & & & & & \\ & 1.00 & & & & & \\ & & 2/\sqrt{5} & 1/\sqrt{5} & & & \\ & & -1/\sqrt{5} & 2/\sqrt{5} & & & \\ & & & & 2/\sqrt{5} & 0 & 1/\sqrt{5} & 0 \\ & & 0 & & 0 & 1.00 & 0 & 0 \\ & & & & -1/\sqrt{5} & 0 & 2/\sqrt{5} & 0 \\ & & & & 0 & 0 & 0 & 1.00 \end{bmatrix}$$

$$\hat{B}^{-1} = \hat{B}^{T}$$

Figure 4. The $\hat{B}$ matrix.

As a second example, suppose that it is desired to approximate the DCT by adding integer products of the WHT vectors. For small integers, this operation can be implemented by digital shifts-and-adds, and requires fewer significant bits to be retained. The matrix $\tilde{C}$, given in figure 5, is an orthonormal transform matrix that is similar to the DCT. The two by two matrix, operating on the vectors of sequency 2 and 6, is a specialization of the general two by two matrix having orthogonal rows with identical factors.

$$\begin{matrix} A & B \\ s_1 B & s_2 A \end{matrix}$$

s_1 and s_2 are plus or minus one, and $s_1 = -s_2$. The four by four operation on the vectors of odd sequency is a specialization of the general four by four matrix with orthogonal rows, identical factors, and the additional requirement of a positive diagonal.

$$\begin{matrix} A & B & C & D \\ -B & A & s_1 D & s_2 C \\ -C & s_2 D & A & s_1 B \\ -D & s_1 C & s_2 B & A \end{matrix}$$

As before, s_1 and s_2 are plus or minus one, and $s_1 = -s_2$.

The specializations of the general matrix were made by requiring that the two by two matrix integers have approximately the ratios found in the second (and third) rows of the $\tilde{A}$ matrix, and that the four by four matrix integers have approximately the ratios found in the fifth (and eighth) rows of the $\tilde{A}$ matrix. Since the $\tilde{A}$ matrix transform is the DCT, this insures that the $\tilde{C}$ transform vectors of sequency 2, 6, 1, and 7 will approximate the corresponding DCT vectors.

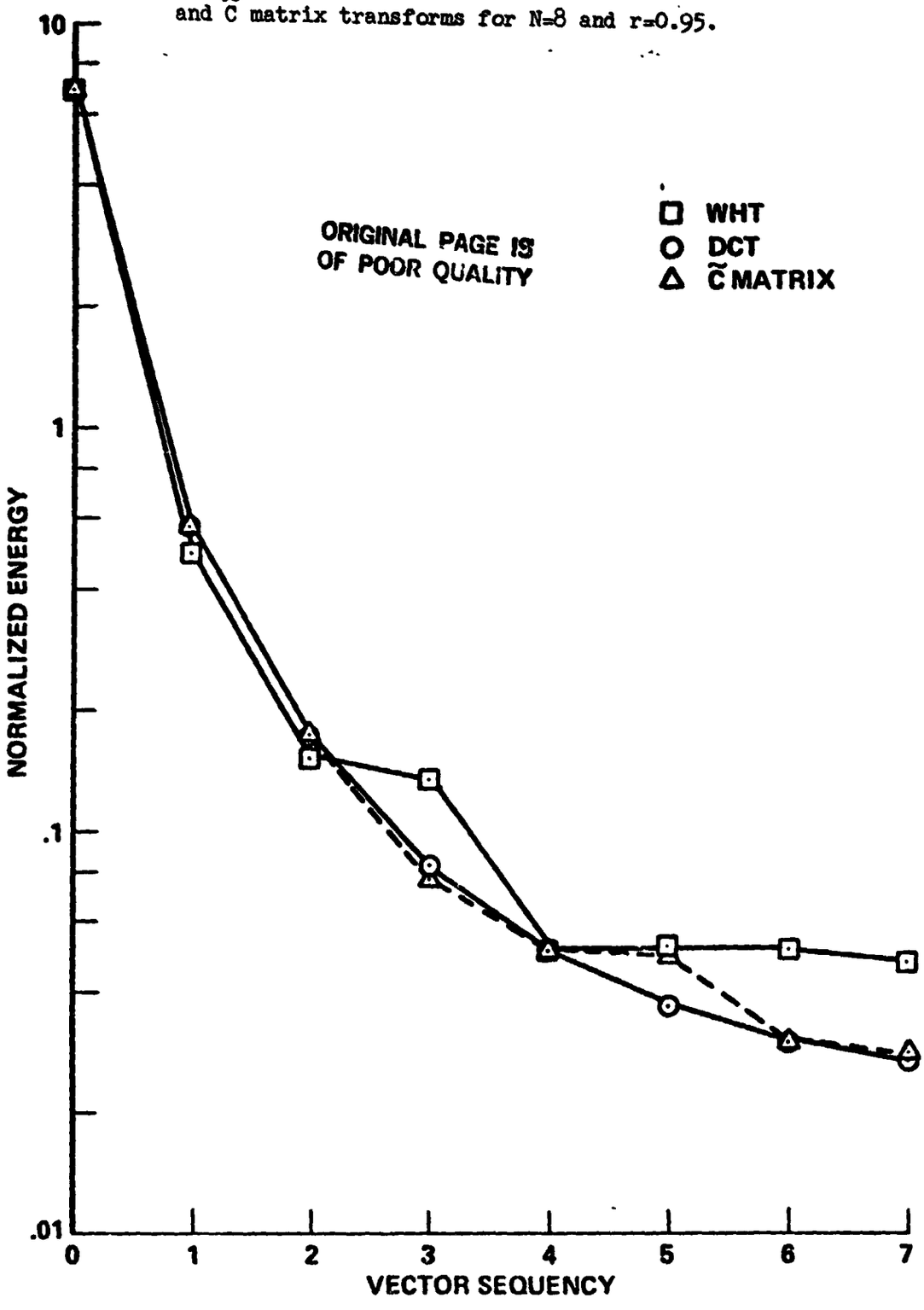
$$\tilde{C} = \left[\begin{array}{cccc|cccc} 13 & & & & & & & \\ & 13 & & & & & & \\ & & 12 & 5 & & & & \\ & & -5 & 12 & & & & \\ \hline & & & & 12 & 0 & 4 & 3 \\ & & & & 0 & 12 & -3 & 4 \\ & & & & -4 & 3 & 12 & 0 \\ & & & & -3 & -4 & 0 & 12 \end{array}\right]$$

$$\tilde{C}\ \tilde{C}^T = 169\ \tilde{I}$$

Figure 5. T . $\tilde{C}$ matrix

The energy compaction results of the C transform, with the results of the WHT and DCT, are given in figure 6, for r = 0.95 and N = 8. The energy of the vectors of sequency 2, 6, 1, and 7 is very similar to the energy of the DCT vectors, but the vectors of sequency 3 and 5 are different. The energy correspondence could be improved by matching the four by four matrix factors to the average of the fifth and sixth rows in the $\tilde{A}$ matrix, but there is little potential data compression remaining. The theoretical performance of the $\tilde{C}$ matrix, in terms of the figure of merit, is given in table II. The $\tilde{C}$ matrix transform obtains nearly all the gain of the DCT over the WHT. If the rational form, instead of the integer form, of the $\tilde{C}$ matrix transform were used, the computation would require sixteen multiplications by four different factors (seven factors including sign differences). There is some reduction in complexity from the implementation of matrix $\tilde{A}$.

Figure 6. The energy compaction of the WHT, DCT, and $\tilde{C}$ matrix transforms for N=8 and r=0.95.



VII. EXPERIMENTAL IMAGE COMPRESSION RESULTS

Experimental results were obtained for two-dimensional, 8X8 sample block implementations of the transforms considered above. Four video test images, Harry Reasoner, Two girls, Two men, and Band, were used in all tests. These images have correlation of 0.97 to 0.98 between elements in the scan line, and fit the first order Markov model, except for the very detailed Band image, which deviates from the Markov model and has an average in-line correlation of 0.85. [11] Two different compression experiments were made.

The test images were first compressed by representing either thirty-two or sixteen of the sixty-four 8X8 transform vectors, using an eight bit uniform, full range quantizer. The other vectors were neglected. The patterns of the vectors transmitted and neglected are given in figure 7. The vectors are in sequency order, with the lowest sequency average vector in the upper left corner of the pattern. The mean-square error for this compression method and the four transforms is given in table III. The $\tilde{B}$ matrix transform error is intermediate between the WHT and DCT errors, and the $\tilde{C}$ matrix error is very close to the DCT error. This is consistant with the Markov model energy compaction results above.

To obtain the greatest transform compression, the transmitted bits should be assigned to the vectors according to equation 1, and the coefficient quantizers should be designed for minimum error given the coefficient energy and amplitude distributions. The optimum theoretical bit assignments and quantizers depend on the particular transform used. The test images, and most typical images, contain low contrast, high correlation background areas, and edges where correlation is low. The bit assignments and quantizer designs based on the stationary Markov model ignore this nonstationarity, and designs which consider low contrast areas and edges give improved mean-square error and subjective performance. Such improved designs have been devised for the WHT, and have been tested with the DCT, $\tilde{B}$ matrix, and $\tilde{C}$ matrix transforms. The transmission rate and mean-square error results are given in figure 8, for the test images compressed in the video field. The DCT gives improved error performance, and the $\tilde{B}$ and $\tilde{C}$ matrix transforms are intermediate, but the $\tilde{B}$ and $\tilde{C}$ matrix results are relatively poorer than in table III. The DCT gives more rate reduction than the WHT, about 0.2 to 0.5 bits per sample. As a two dimensional transform has twice the gain of a one dimensional transform [2],

8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	0	0	0	0
8	8	8	8	0	0	0	0
8	8	0	0	0	0	0	0
8	8	0	0	0	0	0	0
8	8	0	0	0	0	0	0
8	8	0	0	0	0	0	0

32 vector pattern

8	8	8	8	8	8	8	8
8	8	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0

16 vector pattern

Figure 7: Patterns of vector coefficients retained and neglected.

Table III. The mean-square error for the WHT, DCT, $\tilde{B}$ matrix and $\tilde{C}$ matrix transforms with a subset of vectors retained..

Mean-square error for 32 vectors retained

	Reasoner	Two girls	Two men	Band
WHT	0.556	0.806	1.694	3.948
$\tilde{B}$ matrix	0.500	0.738	1.581	3.628
$\tilde{C}$ matrix	0.442	0.666	1.536	3.310
DCT	0.446	0.660	1.535	3.056

Mean-square error for 16 vectors retained

	Reasoner	Two girls	Two men	Band
WHT	1.619	2.206	4.801	12.322
$\tilde{B}$ matrix	1.507	2.093	4.557	12.056
$\tilde{C}$ matrix	1.427	2.029	4.447	11.897
DCT	1.430	2.031	4.406	11.828

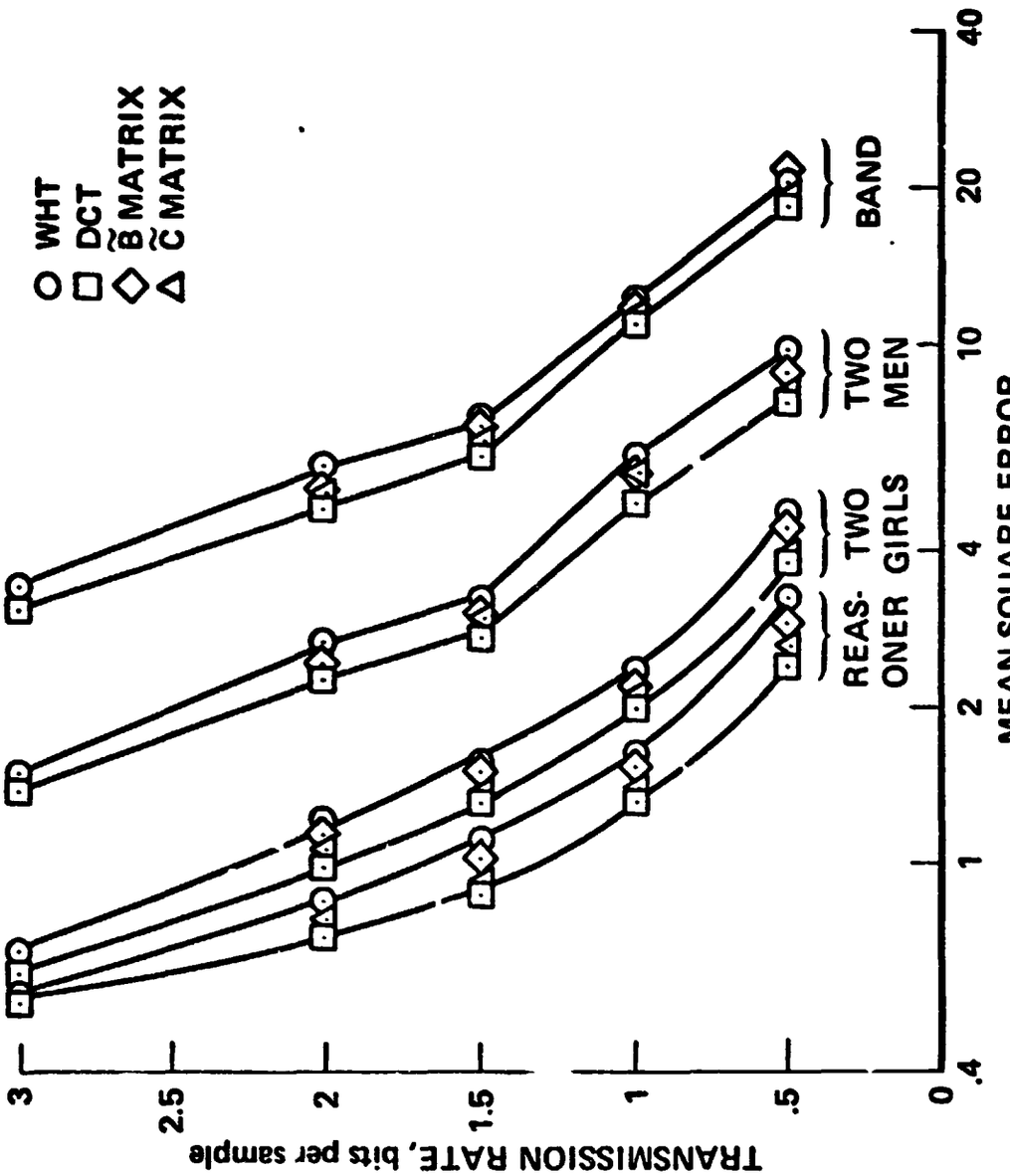


Figure 8. Transmission rate versus error for the four test images.

the theoretical gain of the DCT over the WHT, for r = 0.95, should be twice the 0.15 bits per sample of table II, or 0.30 bits per sample.

The lower error of the DCT, $\tilde{B}$ matrix, and $\tilde{C}$ matrix transforms does indicate subjective improvement in the compressed images. This subjective improvement is larger at lower total bit rates, due to the relative increase of larger, more noticeable errors at the lower rates, and due to the more objectionable, blocky nature of large WHT errors. The $\tilde{B}$ and $\tilde{C}$ matrix errors are subjectively more similar to the DCT errors than to WHT errors, because the higher energy vectors approximate the DCT vectors.

It is not surprising that a design optimized for the WHT gives good results for the DCT and similar transforms. The transform compression introduces errors in three ways; by neglecting vectors,by using too narrow quantizers, and by quantization errors within the quantizer ranges. The DCT, because of its superior energy compaction, reduces the first two sources of error. Although the quantizers used are quasi-uniform, they do have smaller quantization steps for low coefficient values, so the third source of error is also reduced. Any compression design will give better performance with the DCT. From the similarity in energy compaction, a good design for the WHT should be reasonably effective for the DCT. However, further performance gains can be made with the DCT by optimizing the compression designs for the DCT.

The error statistics show that the lower mean-square error of the DCT is due both to fewer large errors, which nearly always occur at edges, and to fewer small errors, which occur in flat areas and edges. The subjective appearance of the compressed image confirms that the DCT produces both smoother low contrast areas and less distorted edges. Since the low contrast areas have very high correlation, and since the edges - though not noise like - can be approximated by a low correlation Markov model, the mean-square error and subjective results agree with the theoretical result that the DCT is superior to the WHT for all values of correlation (see table II).

VIII. CONCLUSION

The Karhunen-Loeve transform for data with stationary correlation and the discrete cosine transform are members of a general class of transforms that can be obtained by a matrix multiplication of the Hadamard vector coefficients. This implementation reduces the number of multiplications required. If reduced compression gain is allowed, the implementation complexity can be further reduced. The theoretical data compression seems to be a reliable indicator of the differential in experimental performance of these transforms.

ACKNOWLEDGEMENTS

The authors are grateful to Drs. D. R. Lumb, S. C. Knauer, and Mr. L. R. Hofman of NASA-Ames Research Center, Moffett field, Ca., and to Dr. N. Ahmed of Kansas State University, Manhattan, Ks., for their assistance and encouragement. The authors are also grateful for the use of the NASA-Ames Research Video Compression Facility.

REFERENCES

1. Ahmed, N., and Rao, K. R., Orthogonal Transforms forDigital Signal Processing, Springer-Verlag, Berlin, 1975.
2. Davisson, L. D., "Rate-Distortion Theory and Application," Proc. IEEE, vol. 60, pp. 800-808, July 1972.
3. Berger, T., Rate Distortion Theory, Prentice Hall, Englewood Cliffs, New Jersey, 1971.
4. Ahmed, N., Natarajan, T., and Rao, K. R., "Discrete Cosine Transform," IEEE Trans. Computers, vol. C-23, pp 90-93, Jan. 1974.
5. Hein, D. and Ahmed, N., "On a Real-Time Walsh-Hadamard/Cosine Transform Image Processor,"
6. Chen, W.-H., Smith, C. H., and Fralick, S. C., "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Trans. Communications, vol. COM-25, pp. 1004-1009, Sept. 1977.
7. Pratt, W. K., Chen, W.-H., and Welch, L. R., "Slant Transform Image Coding," IEEE Trans. Communications, vol. COM-22, pp. 1075-1093, Aug. 1974.
8. Franks, L. E., "A Model for the Random Video Process," Bell Syst. Tech. J., pp. 609-630, April 1966.
9. Connor, D. J., and Limb, J. O., "Properties of Frame-Difference Signals Generated by Moving Images," IEEE Trans. Communications, vol. COM-22, pp. 1564-1575, Oct. 1974.
10. Jain, A. K., "A Fast Karhunen-Loeve Trarsform for a Class of Random Processes," IEEE Trans. Communications, vol. COM-24, pp 1023-1029, Sept. 1976.
11. Jones, H. W., "Theoretically and Experimentally Based Picture Compression Design Methods for Eight by Eight Hadamard Transform Subpictures,"

APPENDIX C

FRAME AVERAGING

This report discusses experiments made in frame averaging. As reported previously, frame repeat has been combined with conditional replenishment, to cope with high change rates. If the fixed average transmission rate of a conditional replenishment system is set at a low value, which would allow compression without repeats for typical teleconference material, the changed frames in high motion scenes can not be fully transmitted in one frame period. Reduced resolution for the changed frame, or updating the changed frame in segments, are less desireable than frame repeat. Frame repeat requires an additional one-field memory at the receiver.

The number of times a frame is displayed (the number of repeats plus one) is shown in Table I, for color and monochrome test scenes processed at different bit rates. The number of times a frame is displayed is equal to the number of frame periods that are required to transmit the next changed frame. Displaying a frame three times, a rate of ten frames per second, is perceptably jerky, but not very objectionable. Although displaying a frame four to eight times is sometimes very jerky, the effect is not intolerable. The subjective effect of frame repeat depends on the kind of motion. In Man and Tool, the original motion is rapid and discontinuous, changing speed and direction. The frame repeat accentuates this. In Cars, the motion in the first part of the scene is smooth and continuous, and the frame repeat at 1/16 bpp remains smooth.

Frame averaging has been used at Bell labs, and was simulated because it was expected to reduce the jerkiness of frame repeat. Frame averaging is practically mandatory when frames of interlaced fields are repeated, because the forward and reverse field display order is otherwise very objectionable. In the current conditional replenishment simulation, changed regions use a field repeat, so that frame averaging is not required.

In the operation of conditional replenishment, a frame is repeated until the next frame is fully transmitted. See the diagram and discussion given in the introductory section above. The number of displays of a frame depends on the amount of change in the next frame, but the amount of change usually varies slowly from frame to frame. After the last portion of a frame has been loaded into the transmitter output buffer, the next frame is input, transformed, and change detected during the next frame period. The number of repeats of a frame can be transmitted

Table I: The number of times a frame is displayed.

	Rate, bpp				
	1	1/2	1/4	1/8	1/16
Scene					
Wheel of Fortune	1	2	3		
Water Skiers	1	2	3-4(3.4)		
Man and Tool			1-2(1.2)	2-4(2.9)	4-8(6.0)
Man and Book				1-3(1.9)	2-5(3.9)
Three People				1	1-2(1.6)
Cars, part 1				2-3(2.2)	3-7(4.4)
Cars, part 2				6-7(6.3)	11-12(11.5)

The number in brackets ( ) is the average.

as soon as the last part of that frame is transmitted. This information is required for frame averaging.

The simulated algorithm for frame averaging uses a gradual mixture of the previous frame (A) and the most recent frame (B). Suppose that the next frame (C) will require four frame periods for transmission. The previous frame (A) has just been displayed in unmixed form. Instead of simply displaying the most recent frame (B) four times, mixtures of the previous (A) and most recent (B) frames will be displayed for three frames, then the unmixed most recent frame (B). If the next transmission also requires more than one frame period, this will be followed by further mixed frames.

Suppose that a sequence of frames require four frame times for transmission. Table II shows the time sequence of input, transmitted, and displayed frames. The expression (2:1,5) designates the mixture of frames 1 and 5 used to replace the untransmitted frame 2 in the display. It seems reasonable that when a frame is closer in sequence to some displayed unmixed frame, it should have more of that frame in its composition. The formula used to define the frame average (I:J,K) is as follows:

$$(I:J,K) = (|I-K|/|J-K|)\ \underline{J} + (|I-J|/|J-K|)\ \underline{K}$$

I, J, and K refer to the frame order numbers, and $\underline{J}$ and $\underline{K}$ refer to the actual frame data. For example:

$$\begin{aligned}
(1:1,5) &= \underline{1} \\
(2:1,5) &= (|2-1|/|1-5|)\ \underline{1} + (|2-1|/|1-5|)\ \underline{5} \\
&= (3/4)\ \underline{1} + (1/4)\ \underline{5} \\
(3:1,5) &= (2/4)\ \underline{1} + (2/4)\ \underline{5} \\
(4:1,5) &= (1/4)\ \underline{1} + (3/4)\ \underline{5} \\
(5:1,5) &= \underline{5}
\end{aligned}$$

The frame displayed in place of the untransmitted frame 2 is an average made by adding 3/4 times frame 1 and 1/4 times frame 5.

Table II: The sequence of input, transmitted, and displayed frames.

Frame order number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Frame input	1				5				9				13	
Frame transmitted		1	1	1	1	5	5	5	5	9	9	9	9	13
Frame repeat display						1	1	1	1	5	5	5	5	9
Frame average display									1	(2:1,5)				
"										(3:1,5)				
"												(4:1,5)	5	

There are several hardware requirements in implementing frame averaging. Knowledge of the number of frame times for the next frame transmission was mentioned above. Frame averaging requires a third one-field memory at the receiver, to hold the previous frame during averaging. Circuitry to combine the frames in different proportions is required. The averaging can be done in the transform domain, using digital circuitry. If both frames were inverse transformed and buffered, they could be combined using analog circuitry.

Frame averaging was tested using conditional replenishment compressed material, consisting of the scenes in table I (except Three People) compressed at the lowest rate shown in the table. The two color scenes, Wheel of Fortune and Water Skiers, were compressed to 1/4 bpp and required three displays of each frame (except the early part of Water Skiers, which required four displays). Frame averaging provided smoother motion, and the detailed moving wheel was improved. While these scenes were not objectionable using field repeat, they were better with frame averaging.

The two averaged monochrome scenes which have been displaved in real time are Mand and Tool and Man and Book, which were compressed to 1/16 bpp and require an average of five displays per frame. These scenes are very jerky using frame repeat, and are not improved using frame averaging. The moving objects - the tool and book- are strangely blurred, and appear to expand and shrink in time. This poor performance is apparently due to both the low frame rate and to the different type of motion. As computed from Table I, the average frame rate for these scenes is six per second, rather than ten per second for the color scenes, and the lowest rate is less than four per second. Both monochrome scenes have a well defined object moving rapidly and discontinuously over a featureless background, and frame averaging causes the shape changing artifact mentioned above. The Cars scene was process 1 with frame averaging, but not displayed. Since the frame repeat scene is smooth, the frame average is expected to be good. The Three People scene was not processed using frame average, as the number of repeats at 1/16 bpp is very low.

Because of the frame averaging artifact described above, the frame averaging was modified to limit the number of averaged frames, and to use repeated frames to fill the remaining display periods. If N is the maximum number of averaged frames,

the expression (I:J,K) is modified as follows:

$$(I:J,K) = (|I-N-2| / |J-N-2|)\ \underline{J} + (|I-J| / |J-N-2|)\ \underline{K}, \text{ for } I \leq N+2$$

$$= \underline{K}, \text{ for } I > N+2$$

The previous example changes as follows for N=2:

$$(1:1,5) = \underline{1}$$
$$(2:1,5) = (2/3)\ \underline{1} + (1/3)\ \underline{5}$$
$$(3:1,5) = (1/3)\ \underline{1} + (2/3)\ \underline{5}$$
$$(4:1,5) = \underline{5}$$
$$(5:1,5) = \underline{5}$$

After frame 5 is received, the display consists of two averaged frames and two repeated frames. For N equal to 2, there are never more than two averaged frames, and the remaining frames displayed are repeats of the most recent frame. The Man and Tool, 1/16 bpp, test scene has been averaged with N equal to one and two, but has not been displayed in real time. It is expected that limited frame averaging will smooth scenes with nigher frame rates, without adding noticeable artifacts where lower frame rates are used. While a single averaged frame probably gives minimal improvement, its implementation does not require the third memory mentioned above.

It appears that frame averaging gives subjective improvement at moderate frame rates, but can not cope with low frame rates. If this is correct, frame averaging is not a highly effective investment in hardware. However, it might be justifiable as an improvement in the limiting mode of a system designed to operate at ten frames per second or more for all change levels. A system using motion tracking might be able to interpolate between frames in a sophisticated way, but would be much more complex.

APPENDIX D

COMPUTER PROGRAMS

AND SIMULATIONS

This report describes the computer work performed under contract NAS2-9703. This work includes the generation of computer programs, the performance of computer simulation test runs, and the transfer of programs and video data to the new SEL 32 computer.

Table I lists the computer programs developed or transferred to the SEL 32. Not all the programs developed under NAS2-9703 have been transferred to the new computer. The function of E88E is included in E88F. IREP and TCOR are all Fortran, and can be easily transferred, but the studies performed using these programs have been completed, reported, and submitted for publication, so that no further need for these programs is anticipated. T8X8 performs cosine and quasi-cosine intraframe transform compression. A cosine transform program is needed, but T8X8 uses 840 machine language subroutines developed by David Heim. E88F simulates the latest conditional replenishment method, and FRAVG generates frame averaged displays. Programs REFMT and ERFMT are derived from a program written by Larry Hofman, which displays on the SEL 32 the 8432 fromat tapes written on the SEL 840. These programs convert the six bit video samples to the most significant six of eight bits, while otherwise retaining the previous format.

In the second part of table I, the list of programs transferred to the SEL 32 includes many frequently used service programs. Notable by their absence are programs to record images (currently impossible), to measure mean-square error (a transferred version of DDIF needs debugging), or to create Dicomed format images from D or E format video tapes.

Table II lists the original video data files transferred to the SEL 32. A few of the files in the sequences D101-D128 and D151-D170 are missing, but all files used in past studies are available on the SEL 32. Sequences ET02, ET04, and ET05 are not transferred because of poor video quality, but ET08 may be useable. All sequences used in previous studies are available on the SEL 32.

Table III lists the conditional simulation test runs made under this contract. Approximately thirty others were made under a previous contract, using an earlier conditional replenishment method. Many of these simulations were recorded on the Echo Science disc, for real time display, but most have been overwritten or have been degraded by time. The Echo Science disc can be readily overwritten from the

Table I. COMPUTER PROGRAMS

Program Function	SEL 840 Name	SEL 32 Name
Programs Developed for NAS2-9703		
Revised Monochrome Conditional Replenishment	E88E	not transferred
Landsat Element Replication Compression	IREP	"
Quasi Cosine Transform Correlation	TCOR	"
Cosine/Quasi-Cosine Compression	T8X8	"
Color/Monochrome Conditional Replenishment	E88F	E88F
Frame Averaging	FRAV	FRAVG
8432 D to SEL 32		REFMT
8432 E to SEL 32		ERFMT
Programs Transferred for NAS2-9703		
Intraframe Hadamard Compression	E8X8	E8X8
Field to Frame (D)	DINT	DINT
Frame to Field (D)	DSTD	DSTD
YIQ to RGB	EYER	ERTOY
RGB to YIQ	DYIQ	
D to E	DTOE	DORE
E to D	ETOD	
D Copy	DCPY	ECOPY
E Copy	ECPY	
D Display	DDSP	DISP4
E Display	EDSP	
Frame 1 Partial Test Ramp	RAMP	JRAMP
Color Test Bars	KCOL	KCOL

Table II. ORIGINAL VIDEO FILES TRANSFERRED

Files	File Description
D101-D128	four frame sequence, monochrome
D151-D170	" " " "
D201-D202	single frame, monochrome
DC01-DC24	single frame, color
LS01-LS04	Landsat frame, D format
ET01	Man and Tool, 85 frame sequence, monochrome
ET03	Cars, " " " "
ET06	Man and Book, " " " "
ET07	Three People, " " " "
EC09,EC10,EC11	Wheel of Fortune, 59 frame sequence, color
EC12,EC13	Water Sking, 46 " " "

Table III. CONDITIONAL REPLENISHMENT SIMULATIONS

Number	Input File	Rate, bits per sample	
Monochrome Simulations Using E88E			
E1	ET01	1/4	
E2	ET01	1/8	
E3	ET07	1/8	
E4	ET06	1/8	
E5	ET03	1/8	
E6	ET01	1/16	
E7	ET01	1/16	
E8	ET06	1/16	
E9	ET07	1/16	
E10	ET03	1/16	
Color Simulations Using E88F (YIQ)			RGB Version
F1	EC09	1/2	
F2	EC09,10, 11	1/2	X
F3	"	1	
F4	EC12,13	1	
F5	"	1/2	X
F6	EC09,10,11	1/4	X
F7	EC12,13	1/4	X

SEL 32 or the SEL 840, but the simulation data has not been converted to the SEL 32 format. The original video sequences, and the conditional replenishment simulation programs have been converted. A monochrome conditional replenishment simulation requires about eight hours, while a color simulation requires about four hours per tape. The simulations of table III required about 140 hours of computer time. Conversion form the SEL 840 to the SEL 32 requires about one hour per tape on each computer, for a total of 25 hours if both computers are run simultaneously. Alternately, the tapes of table III could be converted at the computer center for about 25 dollars each, a total of 625 dollars.

Single compressed images too numerous to list were produced under this contract. They were generated in the design of monochrome and color compression bit assignments and quantizers, in the investigation of Landsat element replication compression, and in the comparison of Hadamard, cosine, and quasiOcosine transforms. Because of the much smaller amount of data, these images could be reproduced or transferred more readily than the conditional replenishment files.

It would be useful to have the SEL 840 available to display previous results, generated in many past studies, but it is far from necessary. An adequate basis for future work exists on the SEL 32.

SECTION C

LANDSAT IMAGE PROCESSING

ELEMENT REPLICATION COMPRESSION

CONTENTS

I. INTRODUCTION

Image transmission rate compression methods either fully preserve the original image information, or introduce errors to some degree. Information preserving compression methods for multispectral images often compute the predicted value of the next image element spectral values, using previously transmitted information, and transmit entropy coded differences between the predicted and actual values.[1]-[3] The transmission rate can closely approach the actual information content of the image. Experiments with Landsat images, which have four spectral bands, have shown that the transmission rate can be reduced from seven or eight bits per element in each band to three or four bits per element in each band.[2], [3] Because of the presence of fine spatial detail, and because the two visible light bands have small correlation with the two infrared bands, Landsat images have higher information content than color television images, and may appear confused to an untrained observer.

Non-information-preserving compression methods remove spatial and spectral information which either is not needed to preserve information or is not useful enough to justify the cost of transmission. Habibi has recently reviewed adaptive compression techniques, and has classified those which introduce error as predictive, transform, or clustering.[4], [5] Predictive techniques can introduce error by transmitting with reduced accuracy, some of the possible differences between an element and the predicted value of the element. Image basis vector transforms in the spatial or spectral domains introduce error if the transform coefficients are transmitted with less than full range or accuracy. When predictive and spatial transform compression techniques are used at rates less than two bits per element per band, fine spatial detail is objectionably blurred.[5], [6], [7]

Clustering is a familiar method of multispectral data classification,[8] and has been combined with multispectral image compression for transmission or storage.[9], [10], [6] The image elements in a region are grouped into four-dimensional spectral clusters, and the four-dimensional centroids of the clusters and the cluster designation of each element are transmitted. Because the errors are spatially uncorrelated with [illegible] there is no spatial blurring in the compressed image. Cluster coded

images have lower error and better subjective performance at low transmission rates than images compressed using predictive or transform techniques. Cluster coding causes errors between the image elements and the centroids used to represent them. These errors are usually invisible in isolated elements, but sometimes cause minor contouring when many adjacent elements have the same centroid. Cluster coding has two disadvantages. Many computations are required to group the image elements into clusters, and to define the centroids. Because of the limited number of clusters, untypical elements may be represented with large error.

Picture element replication has been investigated, to develop a new method spectral, non-spatial, compression without the disadvantages of cluster coding. In replication compression, a table of previously transmitted elements is maintained at both the transmitter and receiver. The distance from the current image element is computed. If a stored element is within the prescribed error distance of the current element, the table indication of that stored element is transmitted, and the current element is represented by the stored element in the receiver image. If many image elements are spectrally similar, the transmission rate approaches the number of bits required to indicate a particular table entry. If no stored element is within the required distance, a special indicator word and the true element value are transmitted.

II. REPLICATION COMPRESSION AND RANDOM CODING

Image compression by element replication is similar to the method of random coding, which achieves the rate-distortion bound.[11]-[13] Proofs of the rate-distortion theorem are related to proofs of the channel capacity theorem.[14]-(16) The rate-distortion proof shows that, as the number of signal dimensions, n, becomes large, a random selection of $M = 2^{nR(d)}$ representing elements will have an element capable of representing any element to be transmitted with distortion less than d, with probability approaching unity. For mean-square error distortion and independant gaussian random variables of variance σ, $d = \sigma^2 \; 2^{-2R(d)}$. Each transmitted element can be indicated by the coded designation of a representing element, at rate R(d), with distortion less than the bound, d. The proof of the converse theorem shows that no possible source coding method improves performance.

The result of the theorem, that there is (with probability approaching one) a representing element with less than the required error, is proved only for large n. Landsat image elements are described using four dimensions, and may have a true dimensionality of two, but even a two dimensional design has higher dimensionality than most designs, which are one dimensional pulse code modulation (PCM). Replication compression uses actual elements, rather than randomly selected elements, as representing elements. This is essentially equivalent to random coding, because the representing elements do not have a simple geometric structure. Signal sets with simply defined geometric structures are inefficient in approaching the random coding bound, because regions centered on such signals have systematic gaps and overlaps in the packing or covering of n dimensional space.[17] New representing elements are added when there is no element within the required error. This corrects for the effect of small n, at the cost of an increase in rate. When a representing element is used, it is indicated by a coded designation, using 4 R bits, of one of the $M = 2^{4R}$ stored elements, where R is the rate per dimension or spectral band.

III. THE ERROR BOUND AND THE EXPECTED ERROR

In element replication compression, an image element is represented by a previously transmitted element only when it is within a predetermined distance of that previous element. The square of the distance bound is the mean-square error bound. Since some replicating elements will be at less than the bounding distance, the expected mean-square error (MSE) is less than the MSE bound.

Suppose that the distance bound is d. After Sommerville [18, p138], the surface content or volume of an n dimensional sphere is

$$S_n = C_n d^{n-1}$$

C_n is a known function of n. The volume of an n dimensional sphere of radius d is

$$V_n = K_n d^n$$

K_n is also a known function of n. After Sommerville, the volume V_n can be obtained by integration over concentric n dimensional surfaces, S_n, as follows;

$$V_n = \int_0^d C_n r^{n-1} dr = \frac{1}{n} C_n d^n$$

$$C_n = n K_n$$

To compute the expected mean-square error (MSE), we assume that the elements represented by a stored element are uniformly distributed over the volume of an n dimensional sphere of radius d, having the representing element as its center.

$$E(MSE) = E(r^2) = \int_0^d (\text{pdf of } r)\, r^2\, (dV_n)$$

$$= \int_0^d \left(\frac{1}{V_n}\right) r^2 C_n r^{n-1} dr$$

$$= (C_n/V_n)\ r^{n+2} / (n+2) \Big|_0^d$$

$$E(MSE) = E(r^2) = \frac{n}{n+2} d^2$$

$$E(MSE) = \quad = \frac{n}{n+2} \text{ MSE Bound} \qquad (1)$$

In table I, the MSE bound and experimental MSE are shown for two images, Salton Sea and Bald Knob. The ratio of the experimental MSE to the MSE bound is about one-half, the calculated ratio for n = 2. The fraction of elements directly transmitted is also shown in the table, and is usually a few percent. Direct transmission causes no error, and the ratio of the experimental error to the bound is corrected accordingly. Because the two visible bands are highly correlated and the two infrared bands are highly correlated, the true dimensionality of the data is more nearly two than four.[5] At small MSE, the dimensionality is higher, because the representing elements have neighbors in all possible directions.

The MSE can be further reduced by using the closest stored element to represent the current element, rather than the first representing element within the required distance. The number of distance computations is approximately doubled. The expected MSE can not be precisely determined, but is estimated in the appendix, under the assumption that the number of alternate representing elements is such that the volume gaps between radius d spheres are equal to the volume overlaps. The resultant ratios of expected MSE to the bound MSE are given in table II, for n = 1,2,3, and 4. In table III, the MSE bound and experimental MSE are given for the case where the closest element in the table is used to represent an element (if the distance is within the bound). The MSE is significantly reduced, from about one-half to about one-third of the MSE bound. The ratios of experimental to bound MSE again are close to the ratios expected for n = 2, the approximate dimensionality of the image data. This result shows that replication compression is efficient in using spectral correlation.

Because replication compression describes a picture element in all spectral bands, the rate, R, is given in bits per element in tables I and III. In sections II and VIII of this paper, the rate is given in bits per element per spectral band, in accordance with general useage. For the Salton Sea image, the tabulated MSE is in units of the

second least significant bit of the original eight bit data. For the Bald Knob image, the data is in units of the least significant bit of the original data. In section VIII, the data plotted for both images is in units of the least significant bit of the original data.

Table I. Mean-Square Error and Error Bounds.
(First Element at Distance $\leq$ d Used)

Salton Sea Image

Table Size in Bits	MSE Bound	Experimental MSE	Exp. MSE / MSE Bnd.	Fraction Transmitted	Corrected Transmission Ratio	Transmission Rate, Two Methods
6	5	2.60	.52	.102	.58	9.05/8.33
5	10	5.01	.50	.053	.53	6.54/5.37
5	15	7.34	.49	.018	.50	5.52/3.95
5	20	9.03	.45	.012	.46	5.34/3.52
4	30	12.88	.43	.014	.44	4.04/2.95
4	40	16.70	.42	.005	.44	4.15/2.57
3	60	24.47	.48	.021	.49	3.56/2.47
		average	.47		.49	

Bald Knob Image

Table Size in Bits	MSE Bound	Experimental MSE	Exp. MSE / MSE Bnd.	Fraction Transmitted	Corrected Transmission Ratio	Transmission Rate, Two Methods
7	5	2.62	.52	.178	.63	12.52/12.75
6	10	5.74	.57	.089	.63	8.66/8.36
6	15	8.61	.57	.042	.59	7.26/6.55
5	20	10.92	.55	.051	.58	6.48/5.95
5	30	15.65	.52	.018	.53	5.53/4.42
4	40	19.87	.50	.034	.52	4.95/4.11
4	60	27.42	.46	.011	.47	4.30/3.20
3	80	31.94	.40	.032	.41	3.87/3.10
		average	.51		.55	

Table II. Theoretical Ratios of Experimental MSE to MSE Bound.

Dimension n	Ratio = n/(n+2) (First Element)	Estimated Change Using Closest	Estimated Ratio Using Closest
1	.33	-.22	.11
2	.50	-.19	.31
3	.60	-.18	.42
4	.66	-.17	.49

Table III. Mean-Square Error and Error Bounds.
(Closest Element at Distance $\leq$ d Used)

Salton Sea Image

Table Size in Bits	MSE Bound	Experimental MSE	Exp. MSE / MSE Bnd.	Fraction Transmitted	Corrected Ratio	Transmission Rate, Two Methods
6	5	2.05	.41	.097	.45	8.91/8.26
5	10	3.84	.38	.051	.40	6.47/5.39
5	15	5.21	.35	.017	.36	5.50/4.07
5	20	6.06	.30	.012	.31	5.36/3.77
4	30	8.75	.29	.013	.30	4.38/2.97
4	40	10.46	.26	.006	.26	4.18/2.64
3	60	18.48	.31	.016	.31	3.42/2.29
		average	.33		.34	

Bald Knob Image

Table Size in Bits	MSE Bound	Experimental MSE	Exp. MSE / MSE Bnd.	Fraction Transmitted	Corrected Ratio	Transmission Rate, Two Methods
7	5	2.00	.40	.171	.48	12.31/12.64
6	10	4.12	.41	.085	.45	8.54/8.41
6	15	5.88	.39	.039	.41	7.18/6.62
5	20	7.17	.36	.050	.38	6.44/5.89
5	30	10.23	.34	.016	.35	5.46/4.62
4	40	11.88	.30	.035	.31	4.97/4.19
4	60	17.19	.29	.013	.29	4.36/3.29
3	80	22.63	.28	.027	.29	3.75/2.92
		average	.35		.37	

IV. THE TABLE SIZE AND THE TRANSMISSION RATE

The number of bits required to indicate one of the representing elements stored in the table is a lower bound on the transmission rate per element. If the number of bits used to indicate a table location is L, and if one location indicator is reserved to indicate direct transmission of an element, the table size is 2^L-1. Suppose that the fraction of elements directly transmitted is f(L), a function of L. If the number of bits needed to directly describe an element is M, and if each directly transmitted element is inserted into the table at a location designated by L bits, the average transmission rate R(L), in bits per element, is

$$R(L) = (1-f(L))\ L + f(L)\ (2L+M)$$

$$= L + f(L)\ (L+M) \qquad (2)$$

The average rate is greater than L by an amount depending on f(L). As L and the table size increase, f(L) decreases. The rate is minimum at some value of L, depending on the image correlation and the MSE bound. The above equation shows that if L is increased by one, rate is reduced if f(L) (L+M) is reduced by more than one, or if f(L) is less by 1/(L+M).

The minimum value of R can be found, if the function f(L) is known. If L is zero, all elements are directly transmitted: f(0) = 1. As L becomes large, repeated transmission of similar elements is avoided, but some minimum number of elements must be transmitted to represent the data within the required error. Suppose that f(L) is larger than this minimum, because of repeated transmissions of similar elements. If L is increased by one, twice as many of the required elements can be stored, and the number of repeated transmissions of required elements can be reduced by one-half. This implies that $f(L) = 2^{-L}$, for f(L) greater than the minimum. f(L) was measured for a wide range of L, at two MSE bounds. The results, plotted in figure 1, are proportional to 2^{-L}, until L becomes large and f(L) becomes small. (The decrease in f(L) for large L indicates that the table management is not optimum.)

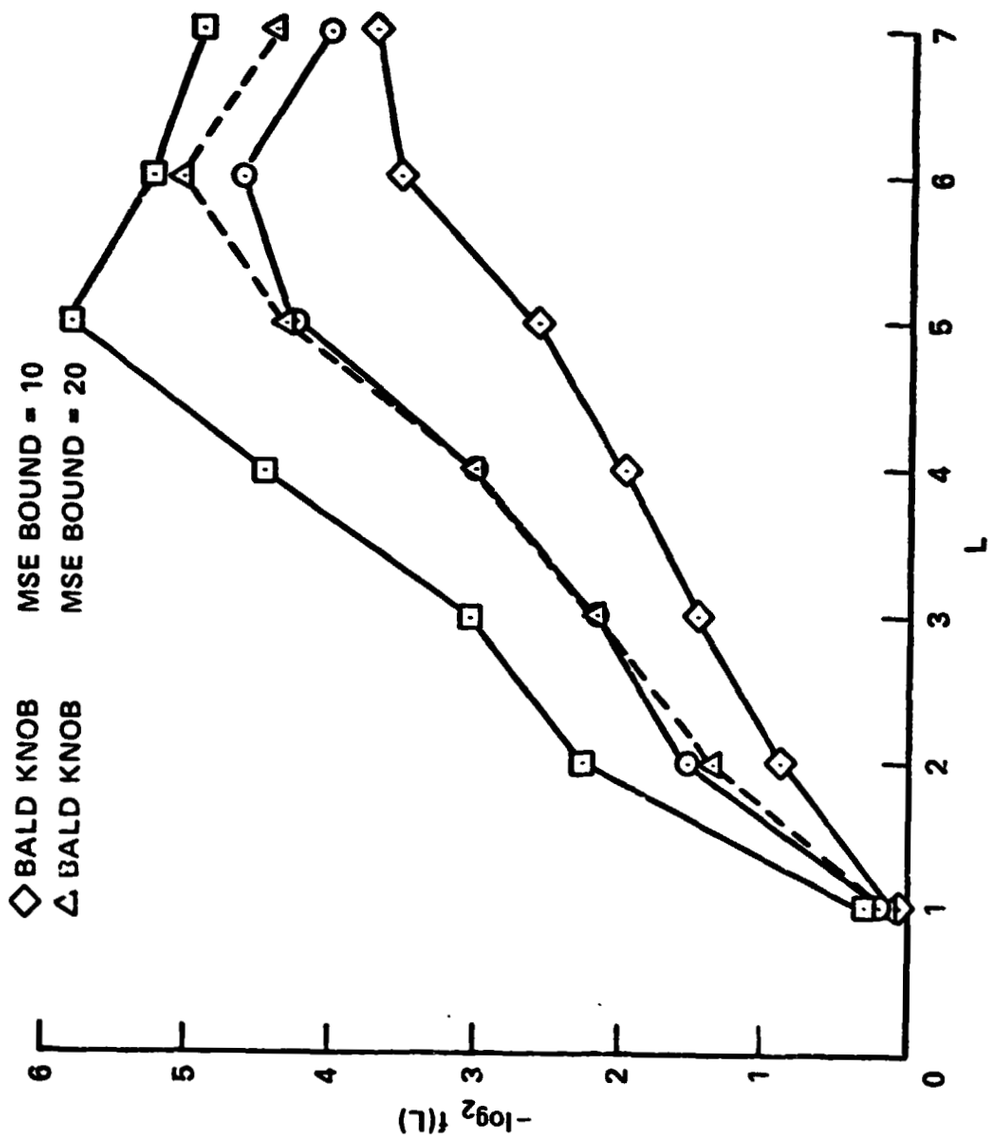


Figure 1: The fraction of elements directly transmitted, $f(L)$, versus the number of bits to indicate table location, L.

We substitute $f(L) = 2^{-L}$ in the equation for R

$$R(L) = L + f(L)\ (L+M)$$
$$= L + 2^{-L}\ (L+M)$$

The minimum rate is found by setting the derivative of R(L) equal to zero.

$$\frac{dR(L)}{dL} = 0 = 1 - 2^{-L}(\ln 2)\ (L+M) + 2^{-L}$$

$$f(L)_{R_{min}} = 2^{-L} = \frac{1}{(\ln 2)(L+M) - 1}$$

$$R(L)_{min} = L + \frac{1}{(\ln 2)(L+M) - 1}\ (L+M)$$

$$= L + \frac{1}{\ln 2 - 1/(L+M)} \qquad (3)$$

$$\approx L + 1.5$$

The first data given in the transmission rate column of tables I and III correspond to the method described above. Data is given only for the L giving the minimum R(L), and R(L) ranges from L + 0.04 to L + 3.05, except for the cases where L equals 7. The larger values of R - L occur for small MSE, where the minimum f(L) is relatively large. The table size could be made adaptive, by increasing L when f(L) is larger than the value for R_{min} given above.

V. THE ERROR BOUND AND THE TABLE SIZE

The table size for minimum rate increases as the error bound decreases. A given distance or mean-square error bound defines an n dimensional sphere, which contains those elements that can be represented by the sphere center. The spheres whose centers are the directly transmitted elements contain all the original data elements. For compression at the minimum transmission rate, a large fraction of the elements to be transmitted must be within the error bound of a currently stored element. If a fixed volume must be filled by spheres, and if the volume of each sphere is decreased by decreasing the MSE bound, the number of the spheres, and the number of centers stored, must be increased.

The volume of the region represented by a transmitted element is

$$V_e = C\, K_n\, d^n = C\, K_n\, (\text{MSE Bound})^{n/2}$$

C is a constant less than one, which allows for the overlap of spheres. For a table indexed using L bits, the total volume represented by stored elements is

$$V_t = 2^L\, C\, K_n\, (\text{MSE Bound})^{n/2}$$

Solving for L,

$$2^L = (V_t / C\, K_L)\, (\text{MSE Bound})^{-n/2}$$

$$L = -(n/2)\, \log_2(\text{MSE Bound}) + \log_2(V_t / C\, K_n) \qquad (4)$$

Since MSE is proportional to the MSE bound,

$$L = -(n/2)\, \log_2 \text{MSE} + \text{a constant}$$

If MSE is reduced by 6 dB, to one-fourth its former value, L must increase n bits, or one bit per dimension. If the effective dimensionality of Landsat data is two, reducing MSE by one-half requires increasing L by one.

The values of L used in tables I and III were selected for minimum rate at each particular error bound. The data of these tables is plotted in figure 2, giving $7-\log_2 MSE$ versus L. The figure is complicated because the same L is optimum for several similar MSE bounds. Solid lines connect the upper points in each set, and dashed lines the lower. The data are in good agreement with the relation found here.

The MSE bound directly determines the MSE, and influences the rate by determining the table size and L for minimum rate, to within a constant.

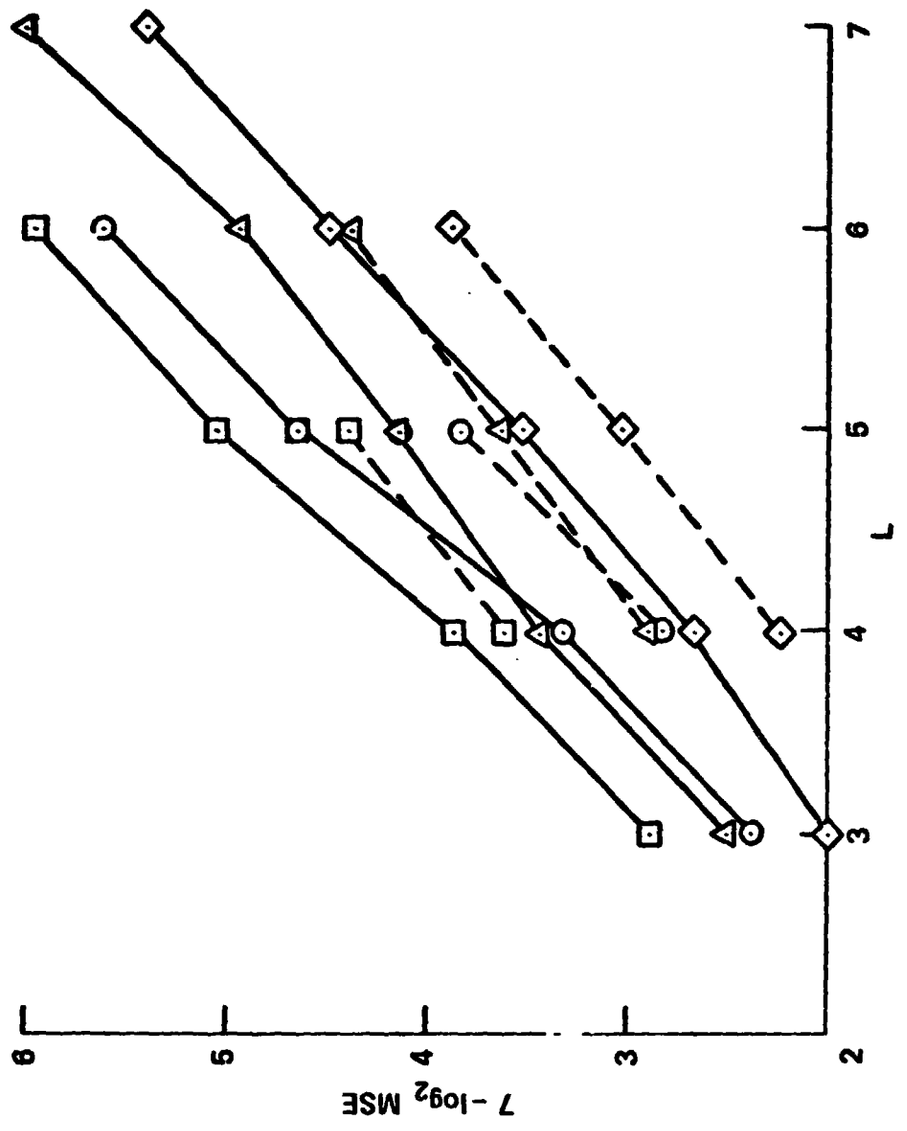


Figure 2: The experimental MSE versus the number of bits to indicate table location, L.

VI. TABLE MANAGEMENT

Some system must be used for deleting and entering new elements in the table of replicating elements. The simplest system is to insert the first transmitted elements into the table in order of transmission, and to replace them in the same order after the table is filled. Although this system is workable, the method adopted was to record the number of uses of each table element, and to replace the least used elements first. The scan method transmits the elements of a 256 element line in order, then begins the next line. The table is sorted according to useage after one-fourth of the table has been replaced, and also at the end of each line of elements. After the sorting at the end of each line, the useage counts are divided by two, to gradually eliminate elements no longer needed. Replacing the table elements according to useage, instead of in arbitrary order, results in small decreases in transmission rate and in MSE, both about ten percent.

In equation 2 of section IV above, 2 L + M bits per element are used when an element is directly transmitted. L bits indicate that the current element is to be directly transmitted, M bits describe the element, and a second L bits indicate the table location where the transmitted element is to be stored. If the algorithm determining the table location is applied at the receiver as well as at the transmitter, the L bits defining the element to be replaced are not required. However, the average rate is not much affected by this change. Equation 2 becomes

$$R(L) = L + f(L)\ M$$

Equation 3 becomes

$$R(L)_{min} = L + 1/\ln 2$$

The experimental rates were obtained using 2 L + M direct transmission bits, but the information in tables I and III is sufficient to allow calculation of the rate for L + M direct transmission bits.

It was noted in section IV that the decreasing f(L) for large L, shown in figure 1, indicates that the table management is not optimum. If the most useful $2^L - 1$ elements were always retained in the table, f(L) would always decrease as L increases. However, it is difficult to select correctly the elements that will be needed to replicate future input elements. Perhaps the ordering of elements according to useage should be more frequent for larger L. When table size is large, it may happen that several stored elements are used to represent elements in a region that one element would represent in a small table. As the useage counts for the several elements are smaller, they all may be replaced when one stored element with the total useage count would be retained. Since f(L) is quite small for large L, little rate reduction can be gained by improving table management, and no further experiments were made.

VII. COMPRESSION USING SPATIAL CORRELATION

Element replication compression is primarily spectral, but the elements contained in the table change as the image is scanned. Any spatial proximity of similar elements increases the probability that a useable replication element will be in the table. It is also possible to indicate recently used elements by a few bits, so that spatial correlation would reduce the average transmission rate. The simplest method, which is used here, is to use a one bit word to indicate that a replicating element is the same as the last one used, and to prefix all the L bit table designations by the complementary one bit word.

If the probability that the previous replicating element is used again is p, the rate for this method is

$$R_1 = p + (1-p-f(L))(L+1) + f(L)\,(2L+1+M)$$

The rate for the previously described system is, from equation 2,

$$R = (1-f(L))\,L + f(L)\,(2L+M)$$

$$R-R_1 = -p + pL - (1-p-f(L)) - f(L)$$

$$= pL - 1$$

The new method achieves a reduction in rate if $pL > 1$. At the lower transmission rates, L is 3, 4, or 5, and p is 1/2 or more, so that the transmission rate is reduced about one bit per element. Data obtained using this method is given in tables I and III, as the second number in the transmission rate column. The gain for large table sizes is small, since p is small.

If a one bit word indicates the last table entry used, and a two bit word indicates some other entry with probability q of being used, the L bit table designations must be prefixed by the complementary two bit word. The rate for the two indicator method is

$$R_2 = p + 2q + (1-p-q-f(L))(L+2) + f(L)(2L+2+M)$$

$$R_1 - R_2 = -2q + q(L+1) - (1-p-q-f(L)) - f(L)$$
$$= q\,L - 1 + p$$

There is a further reduction in rate due to the second special designation, if $q > (1-p)/L$. For $L = 5$ and $p = 1/2$, q must be greater than 1/10; if q is 1/5, the rate reduction is 1/2 bit per element. No experiments were made using such second special designations.

VIII. IMPLEMENTATION AND RELATIVE PERFORMANCE OF REPLICATION COMPRESSION

Multispectral image compression was implemented as a computer program, in the following manner;

1) The mean-square error bound is selected.

2) The stored element table size, $M = 2^L - 1$ is selected. At the beginning of each image, the table contains all zero elements.

3) For each element to be transmitted, distance was measured to all the elements stored in the table.

4a) If a stored element is within the MSE bound, the table designation of the closest element is transmitted, using L bits.

4b) Alternately, if the table designation is the same as that last used, a one bit word is transmitted. If the table designation is different, the correct L bit table designation, prefixed by the complementary one bit word, is transmitted.

5) If no stored element is within the MSE bound, the reserved indicator word of L bits (or L+1 bits per 4b) is transmitted, followed by M bits describing the element, and L bits indicating the table position for its storage.

6) After one-quarter of the table elements have been replaced, and also at the end of each line, the stored elements are ordered according to useage. In 5, the least used will be replaced first. The useage counts are reduced by one-half, after each end-of-line ordering, to reduce the effect of spatially distant elements.

The rate and MSE results for two images have been given in tables I and III. In table I, the first stored element at distance less than the bound was used to represent the compressed element. In table III, the closest stored element at distance less than the bound was used. In both tables, the first number given for transmission rate is for the method not using a special repeat designation (4a), and the second rate number is for the method using the repeat designation (4b). The rate and MSE data of table III is plotted in figures 3 and 4. The plotted rate is in bits per element per spectral band. The mean-square error is given in units per band, where an error of one corresponds to an error in the least significant bit of the eight bit data. Previous results from several sources are shown for comparison.

Figure 3: Rate versus distortion for the Salton Sea image.

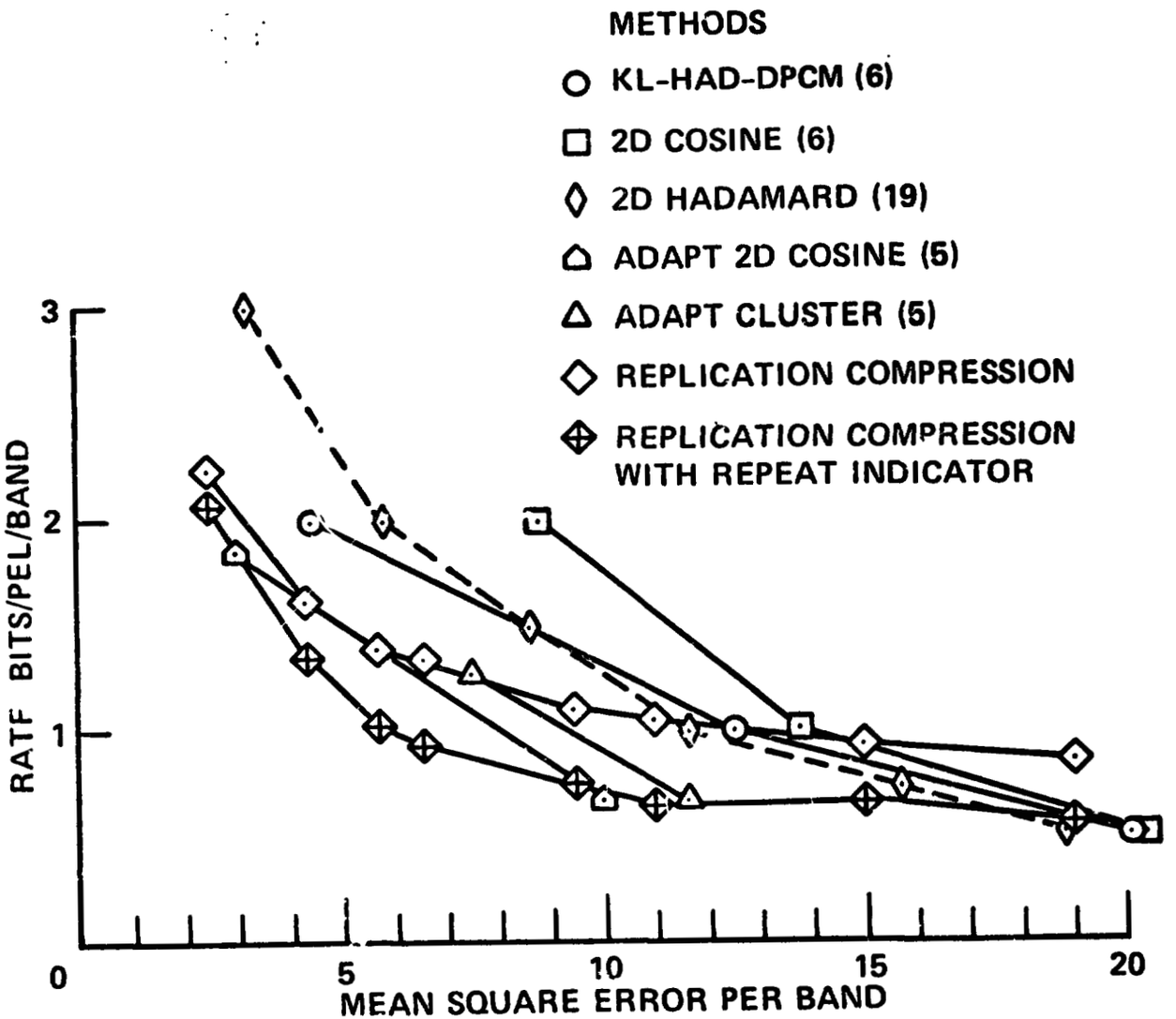
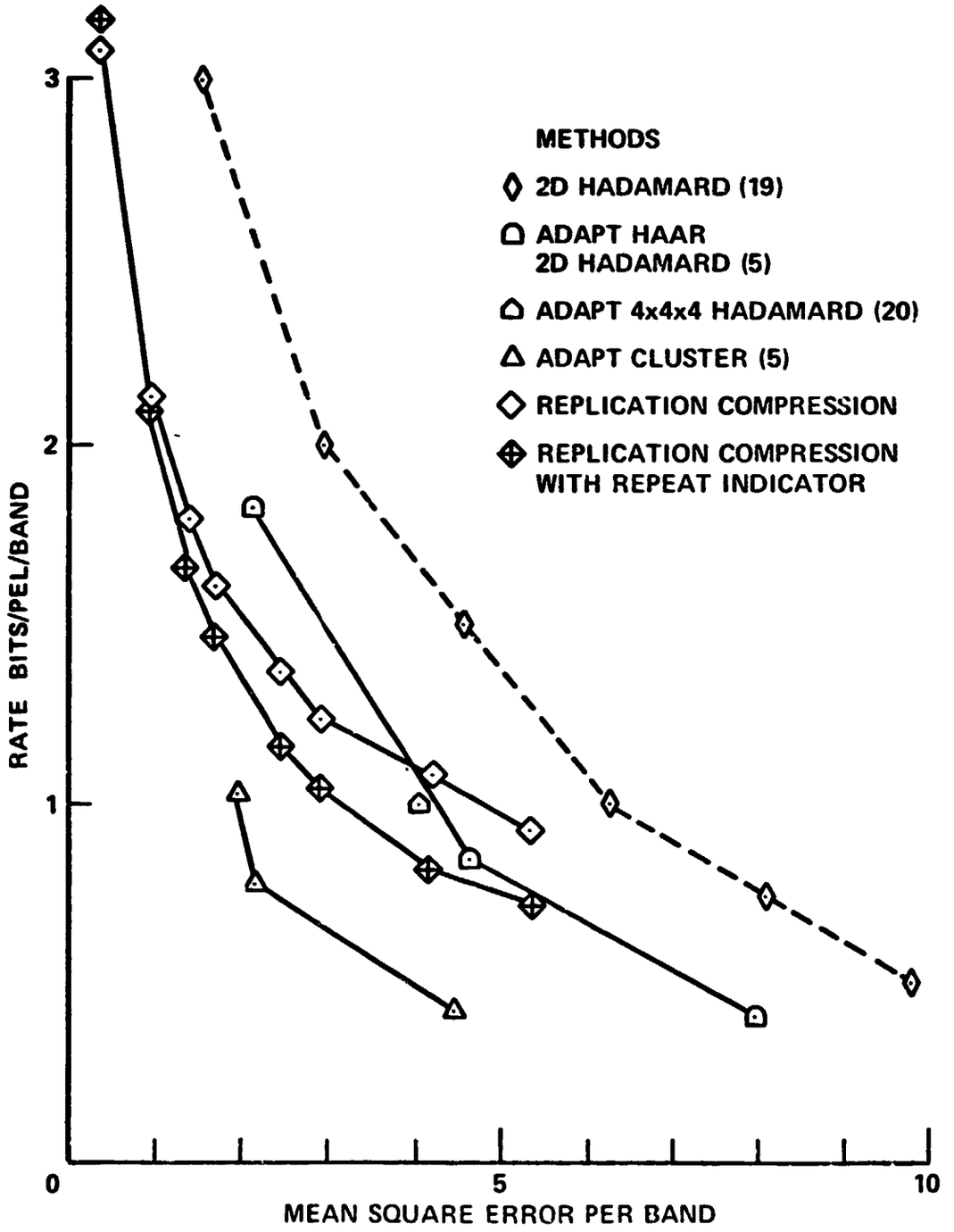


Figure 4: Rate versus distortion for the Bald Knob image.



For the Salton Sea image, figure 3, the results of two dimensional cosine[6] and two dimensional Hadamard transform compression [19] are shown. Each band is processed independantly. The KL-Hadamard-DPCM compression uses a Karhunen-Loeve spectral transform, followed by a horizontal Hadamard transform and vertical differencing in each transformed band.[6] These three methods use a fixed rate for each element; the other methods shown all have varying rate. The adaptive cosine transform compression is two dimensional spatial compression, using different transform vector bit assignments and quantizations in regions of different spatial detail.[5] The adaptive cluster coding groups data elements into spectrally similar clusters, merges clusters that have centroids within a predefined distance, and entropy codes the class designations.[5] For the Salton Sea image, replication compression and adaptive cosine compression give the best results.

For the Bald Knob image, figure 4, two dimensional Hadamard compression is the only non-adaptive technique. Two adaptive three dimensional transform techniques were used with similar results. In the adaptive Haar Hadamard method, a Haar transform in the spectral domain was followed by an adaptive, variable rate two dimensional Hadamard transform.[5] The three dimensional Hadamard transform adapted to spectral differences, using a fixed rate.[20] For the Bald Knob image, adaptive cluster coding gave the best results, followed by replication compression with a repeat indicator.

Figure 5 shows the originals of bands 1 and 4 of the Salton Sea image, and the same compressed using replication compression. This compression test used L equal to 5. and an MSE bound of 10. Using the closest stored element and a repeat indicator, the transmission rate is 5.39 bits per element, or 1.35 bits per element per band. The average MSE per band, in units of the least significant bit of the original eight bit data, is 4.34.

Figure 6 shows original and compressed bands 1 and 4 of the Bald Knob image. For this compression test, L equalled 5, and the MSE bound was 20. Using the closest stored element and the repeat indicator, the rate is 5.89 bits per element, or 1.47 bits per element per band. The average MSE per band, in terms of the least significant bit of the original data, is 1.79.

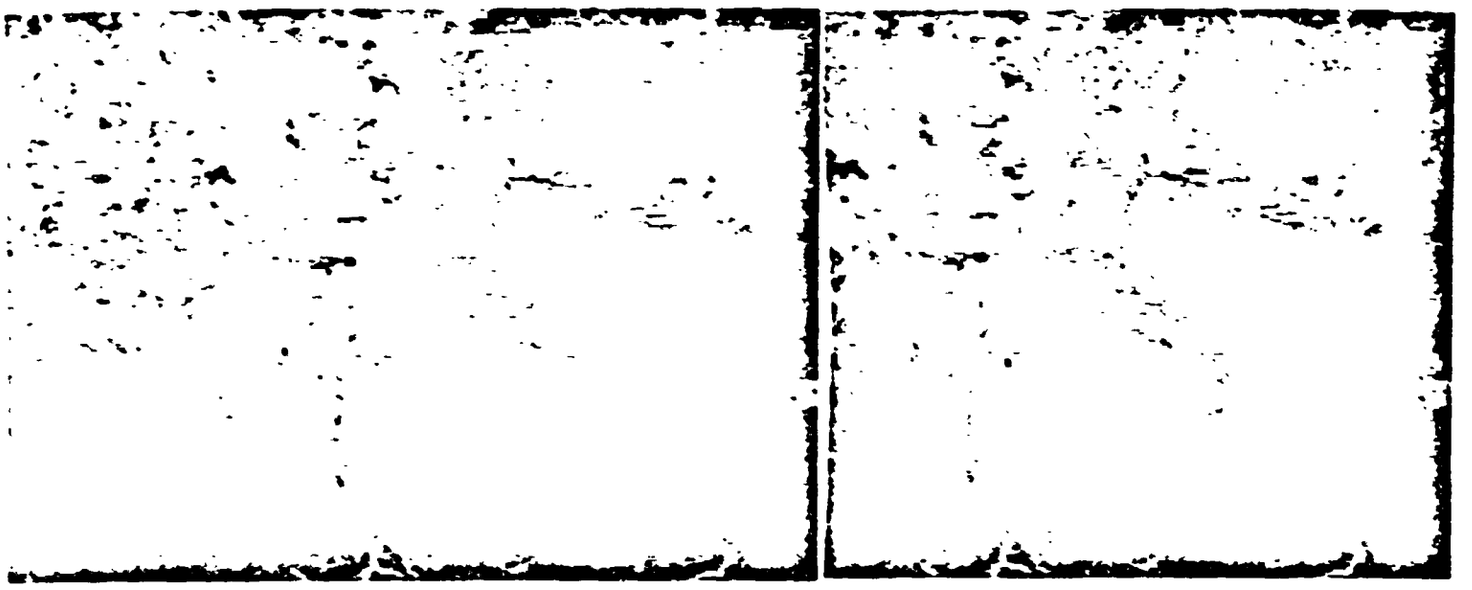
Original, band 1

Compressed, band 1

Original, band 4

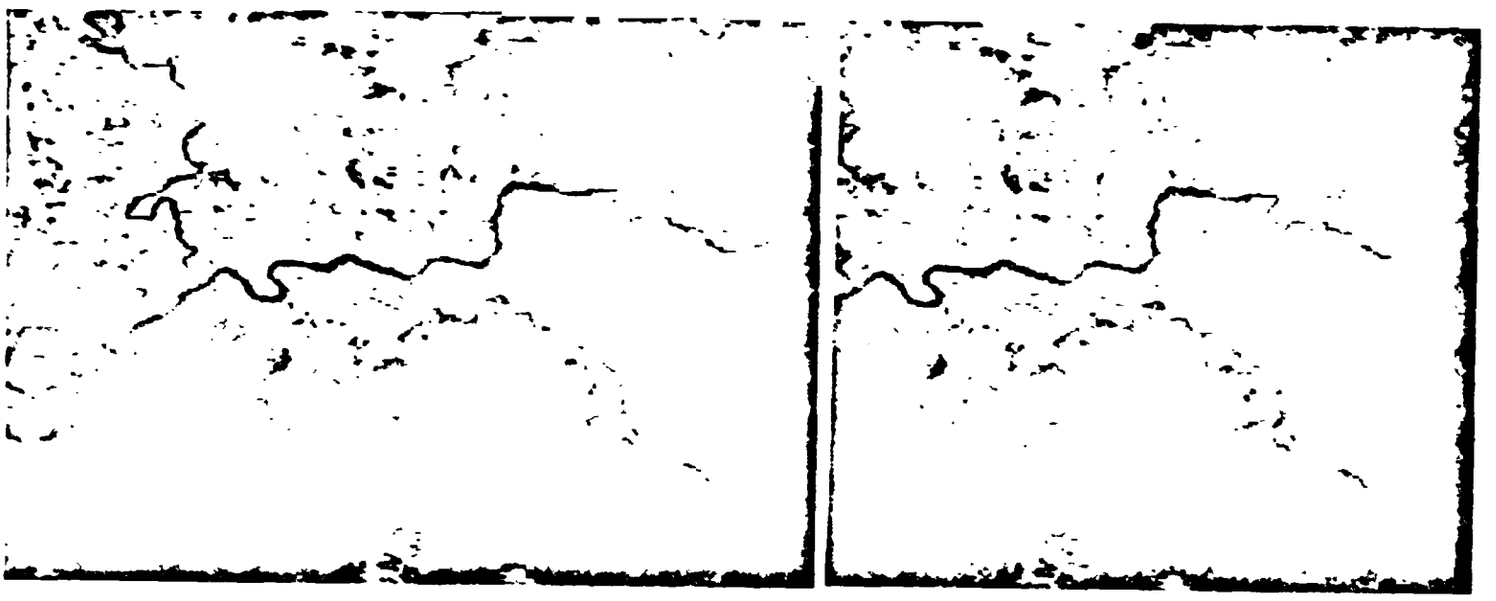
Compressed, band 4

Figure 5: Salton Sea image, original and compressed bands 1 and 4, 1.35 bits per element per band and 4.34 average MSE per band.



Original, band 1

Compressed, band 1



Original, band 4

Compressed, band 4

Figure 6: Bald Knob image, original and compressed bands 1 and 4, 1.47 bits per element per band and 1.79 average MSE per band.

The Salton Sea image has 240 elements and 248 horizontal lines. The Bald Knob image has 176 elements and 256 lines. The individual picture elements are visible in the images, because each element was duplicated six times in both the vertical and horizontal directions. The method of replication compression preserves high contrast spatial detail, but introduces false contouring in low contrast areas.

XI. REPLICATION COMPRESSION AND CLUSTER CODING

The above results show that selective replication, with a repeat indicator, gives performance comparable to cluster coding. Both methods use spectral compression and a table entry indication. The two techniques will be compared in computation requirement, maximum error, subjective image appearance, and spatial compression characteristics.

In replication compression, each input element requires one distance computation for each stored table element. In cluster coding, the initial clusters are formed by computing the distances between elements and joining the closest. The centroids of the clusters are computed, and the distances between them are computed to determine which clusters can be merged. To reduce the computation requirement, clusters are formed over two dimensional subpictures, which were 16 by 16 for the results referred to above. The 256 elements have about 128 distances each, which corresponds to the distance computation for replication compression using a seven bit table. The typical table size of five bits requires one-quarter as much computation. After the initial distance computation, cluster coding requires many more computations, depending on the image data.

In replication compression, the maximum error is defined by an input parameter, and is typically three times the average error. In cluster coding, the limited number of clusters may cause large error. Suppose we are restricted to sixteen initial classes in four dimensions. The sixteen centroids with maximum separation in four dimensions are the vertices of the four dimensional hypercube. These are at $(\pm A,\pm A,\pm A,\pm A)$, where A is the maximum amplitude in each dimension. Suppose an element occurs at the center (0,0,0,0), and must be merged with a cluster centered at one of the vertices. If there are many elements in that cluster, the new point will change the centroid little, and the error in representing the center point is $4\ A^2$. This error is equal to the square of the largest possible element value. This example is extreme, and most errors are small, but the maximum error in cluster coding is data dependant, and may be very large.

As cluster coding and replication compression are both spectral methods, the subjective appearance of the compressed images is similar, and avoids the spatial

blurring introduced by predictive and transform compression at comparable rates. However, because cluster coding uses two dimensional subpictures, the subpicture edges are visible in images compressed at low rates [5]

Replication compression makes use of spatial correlation by placing current elements in the stored table, and by indicating repeated replication elements by a one bit word. Adaptive clustering makes use of spatial correlation by forming local clusters, which are few (reducing rate) and spectrally similar (reducing error) when spatial correlation is high. Because of the variable number of clusters, cluster coding can achieve a low rate in areas of high correlation. Because of the entropy coding of cluster indicators, the occurance of a few unusual elements does not significantly increase rate. Replication compression could be similarly improved by making the list size adaptive and by entropy coding the list indicators. The replication algorithm used here uses only past elements and the current element, and the gain of adaptive list size and entropy coding would be increased if the replication algorithm examined future elements before compression, as does cluster coding.

X. CONCLUSION

A method has been developed for multispectral image compression, using replication of stored elements with an error bound. The method has been theoretically described, and experimental performance has been compared with other compression techniques. The average mean-square error for two Landsat images is about one-third of the error bound, as expected from theoretical considerations. For $2^L - 1$ stored elements, and for the L giving minimum rate, the transmission rate is about L + 1.5. To reduce the mean-square error to one-fourth its value, L and the transmission rate per element must increase about one bit per effective dimension. Performance is improved slightly by having the stored elements include more frequently used and more recently used elements. Having a one word indicator for a repeated use of the same element gives significant rate reduction. The rate and mean-square error performance of replication compression is superior to that of most previously examined techniques, and is similar to that of adaptive cluster coding. The implementation of replication compression is simple, and is similar to the random coding, table look-up method shown to achieve the rate-distortion bound.

APPENDIX. EXPECTED MEAN-SQUARE ERROR WHEN THE CLOSEST STORED ELEMENT IS USED

Mean-square error can be found by integration over the appropriate n dimensional volume. Initially, we consider the case where the distance bound is d, and there are two elements at distance $2b < 2d$, as shown in figure A.1. The expected MSE contribution to the left hand sphere, due to the shaded cap, is

$$E(MSE)_l = \int_b^d 2 \int_0^{(d^2-z^2)^{1/2}} \int_0^{(d^2-y^2-z^2)^{1/2}} (1/V_n)\ (r^2+y^2+z^2)(C_{n-2}r^{n-3}\,dr\,dy\,dz)$$

The three terms in the integral are the probability density function of r, the measure of squared error, and the volume element. Using the closest element transfers the points in the shaded cap to the right hand element, and the expected MSE is

$$E(MSE)_r = \int_b^d 2 \int_0^{(d^2-z^2)^{1/2}} \int_0^{(d^2-y^2-z^2)^{1/2}} (1/V_n)\ (r^2+y^2+(2b-z)^2)(C_{n-2}r^{n-3}\,dr\,dy\,dz)$$

$$E(MSE)_{change} = E(MSE)_r - E(MSE)_l$$

$$= \int_b^d 2 \int_0^{(d^2-z^2)^{1/2}} \int_0^{(d^2-y^2-z^2)^{1/2}} (1/V_n)\ 4b(b-z)\ (C_{n-2}r^{n-3}\,dr\,dy\,dz)$$

After integration using $y = (d^2-z^2)^{1/2} \sin s$, and $z = d \cos t$,

$$E(MSE)_{change} = \int_0^{\cos^{-1}(b/d)} (4b^2K_{n-1}/K_n)\ \sin^n t\ dt - \frac{4bK_{n-1}\ (d^2-b^2)^{(n+1)/2}}{K_n\ d^2\ (n+1)}$$

$$= F(n,b,d)$$

For small n, this equation is readily integrated.

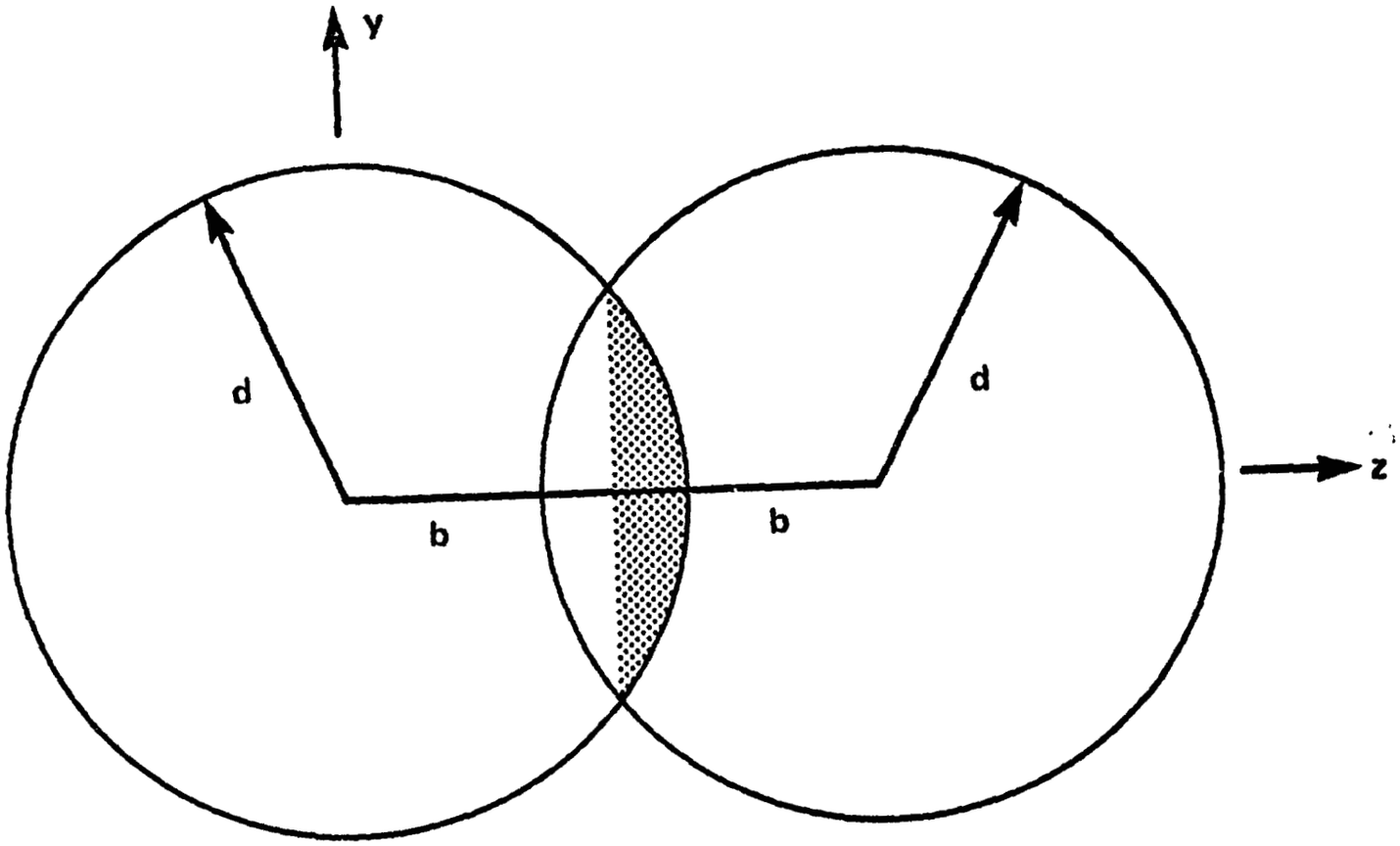


Figure A 1. Two spheres at center distance 2b > 2 d, and the volume affected by using the closest center.

We have found the MSE reduction in one element representing region, due to one neighboring sphere at distance 2b. The expected MSE reduction for an element representing region depends on the number of neighboring elements at different distances. To estimate this number, we assume that the volume of neighboring spheres exactly equals the local volume available, including the volume of the original sphere. That is, the volume of the gaps between spheres is assumed equal to the overlapping sphere volume. As shown in figure A.2, an element at distance 2b from another element has more than one-half its volume within distance $(4b^2+d^2)^{1/2}$ of that element. The number of spheres at distance 2b is then

$$N(b) = \frac{V_n (4b^2+d^2)^{n/2}}{V_n d^n}$$

$$= (4b^2/d^2 + 1)^{n/2}$$

The expected MSE is then

$$E(MSE) = \int_0^d F(n,b,d) \; N(b) \; db$$

The values obtained from this integration are given in table II, in the text, for dimension n = 1,2,3, and 4.

ACKNOWLEDGEMENT

The author is grateful to NASA Ames Research Center for financial support, for the opportunity to use the Ames Image Compression Facility, and for valuable discussions with research scientists there. The author is grateful to Dr. Ali Habibi of the TRW Defense and Space Systems Group, and to Dr. Nasir Ahmed of Kansas State University for permission to use and describe their experimental results.

REFERENCES

(1) R. F. Rice and J. R. Plaunt, "Adaptive Variable Length Coding for Efficient Compression of Spacecraft Television Data," IEEE Trans Com. Tech., vol. COM-19, pp. 889-897, Dec. 1971.

(2) C. L. May and D. J. Spencer, "ERTS Image Data Compression Technique Evaluation," TRW Defense and Space Systems Group, NASA Contract NAS5-21746, April 1974.

(3) P. Capitant, "Interactive Link Compression Technique," Stanford University, Final Report, NASA Interchange Agreement NCA2-OR745-614, Sept. 1

(4) A. Habibi , A.S.Samulon, G.L.Fultz,and D.R.Lumb, "Adaptive Coding GS Imagery," NTC'77 Conference Record, vol. I, pp 10:2-1 to 10:2-8, Dec. 1977.

(5) A. Habibi andA.S. Samulon, "Study of Adaptive Techniques for Data Compression of Multispectral Scanner Data," TRW Defense and Space Systems Group, NASA Contract NAS2-8394, March 1977.

(6) A. Habibi, "Study of On-Board Compression of Earth Resources Data, TRW Systems Group, NASA Contract NAS2-8394, Sept. 1975.

(7) A. Habibi and A.S.Samulon, "Bandwidth Compression of Multispectral Data," Proceedings of SPIE, vol. 66, pp. 23-35, August 1975.

(8) G. Nagy, "Digital Image-Processing Activities in Remote Sensing for Earth Resources," Proc. IEEE, vol. 60, pp. 1177-1200, Oct. 1972.

(9) E. E. Hilbert, "Joint Classification and Data Compression of Multidimensional Information source - Application to ERTS," ICC'75, Conference Record, vol. II, pp. 27-1 to 27-6.

(10) E. E. Hilbert, "Joint Pattern Recognition Data Compression Concept for ERTS Multispectral Imaging," Proceedings of SPIE, vol.66, pp. 122-137, Aug. 1975.

(11) C. E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion," IRE Nat. Conv. Rec, pt. 4, pp.142-163, Mar. 1959.

(12) D. J. Sakrison, Notes on Analog Communication, New York, Van Nostrand Reinhold, 1970, ch. 6.

(13) T. Berger, Rate Distortion Theory: A Mathematical Basis f Data Compression, Englewood Cliffs, NJ, Prentice-Hall, 1971, pp. 66-75.

(14) C. E. Shannon, "A Mathematical theory of Communicat ," Bell Syst. Tech. J., vol. 27, pp. 379-423, July 1948.

(15) C. E. Shannon, "Communication in the Presence of Noise," Proc. IRE, vol. 37, pp. 10-21, Jan. 1949.

(16) J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York, Wiley, 1965, pp. 320-341.

(17) H. W. Jones, Geometric Capacity Bounds and Signal Designs, Ph. D. Dissertation, University of Pennsylvania, 1974. University Microfilms, Ann Arbor, Mich.

(18) D. M. Y. Sommerville, An Introduction to the Geometry of N Dimensions, New York, Dover 1958, pp. 135-140.

(19) H. W. Jones, "Theoretically and Experimentally Based Picture Compression Design Methods for Eight by Eight Hadamard Transform Subpictures," to be published.

(20) N. Ahmed and T. Natarajan, "An Adaptive Transform Coding Approach for Multispectral Scanner Data, Kansas State University, NASA Interchange NCA2-OR363-601, 1976.

APPENDIX A

A BRIEF REVIEW OF

LANDSAT IMAGE CLASSIFICATION

INTRODUCTION

This report describes the current state of Landsat data classification by computer, and considers possible areas of research for NASA-ARC. The fundamental problem is obtaining useful discrimination of various phenomena, given the spectral, spatial, and temporal characteristics of Landsat data. Improved accuracy sometimes justifies the cost of more sophisticated computer processing, and different classification methods provide alternate approaches for special problems.

LANDSAT IMAGE SYSTEMS

The classification of Landsat images is one of many operations performed by a computer image processing system. JPL's VICAR, ESL's IDIMS, GE's Image 100, and Purdue's LARS are well known. The functions usually performed by these systems are as follows; 1-input (tape to disc, reformat, data check), 2-mosaic (merge image files), 3-clean up (remove bad lines, banding), 4-statistics (histograms, ratios, classifier parameters), 5-enhancement (smoothing, edge emphasis), 6-display (false color, area enlargement, contrast stretch), 7-clustering (classification centers), 8-classification (training samples, maximum liklihood, statistics), 9- mapping (cross correlation, overlay, registration), 10- output (data tape, photo copy). The most direct way to investigate classification methods using the SEL 32, would be to obtain IDIMS or EDITOR format, cleaned up, 800 cpi data tapes, and program the SEL 32 to display, cluster, and classify such input data. If useful classifiers are developed, their results could be made compatible with the IDIMS or EDITOR output functions. The alternate approach would be to develop a complete image system for the SEL 32, which would require extensive time and effort.

FAMILIAR LANDSAT DATA CLASSIFIERS

The most widely used Landsat classifier assumes that each class has a normal multivariate distribution, and uses the maximum liklihood decision rule. The training samples for each class are described by a vector of four spectral values. The training samples are used to compute the mean and variance of each class, which completely define the normal multivariate distribution. For each unknown sample, the probability that the sample is a member of each class is computed from the distribution and the

a priori class probability, and the class with the highest probability is selected.

This classifier requires the computation of quadratic forms. Classification of a full Landsat frame requires about 90 hours on the IDIMS HP-3000, 2 hours on an array processor, and about 20 minutes on the ILLIAC IV. Several methods have been used to increase the speed of the normal distribution, maximum liklihood classifier, including table look up (11, 17 p370, 19 p755), sequential decision trees (19 p778), linear approximation to the quadratic discriminant (19 p780), and canonical sucessive approximations to the quadratic discriminant (7).

If all the classes have the same form of distribution (ie., the same a priori probability and covariance matrix) a maximum liklihood classifier uses a linear rather than a quadratic discriminant (5 p29). If the four spectral values are also independant and have equal variances, the correct maximum liklihood decision rule is to assign a sample to the class having the closest mean or centroid vector (5 p27). These classifiers are faster than the classifier for the general normal multivariate distribution, but they are usually less accurate, because their assumptions are less realistic. Because these classifiers assume the form of the class distribution, and compute the distribution parameters from the training samples, they are termed parametric classifiers.

Clustering is similar to classification, but uses a different approach. Maximum liklihood classification is supervised, by using training samples. Clustering is unsupervised, and examines only the data. Clustering methods define the inherent spectral data groupings, using distance measures and group combining and splitting. If the data clusters are assigned to classes, the process is similar to minimum distance to centroid classification, which is less accurate than normal distribution maximum liklihood classification.

It is obvious that multimodal class distributions will cause errors to be made by the normal distribution maximum liklihood classifier. Clustering is usually used to split the classes into spectrally homogeneous and separable subclasses, which can be used to train the normal distribution maximum liklihood classifier. After classification, the subclasses are recombined into the original classes.

NONPARAMETRIC CLASSIFIERS

Because the multispectral data does not fit the normal distribution model (19 p780), nonparametric classifiers are sometimes used. A direct approach to nonparametric classification is to use the training samples to estimate the class probability distributions, and to perform maximum liklihood classification using these empirical distributions. Density estimation requires more computation than normal maximum liklihood classification, but some reductions are possible (19 p778). One experiment found that estimated probability distributions gave no significant improvement in classification accuracy (19 p780); another test found classification accuracy improved from 95.1% to 100% using four features (10, 3 p156-162, 19 p784).

The nearest neighbor classifier is an important nonparametric method. All the training samples are stored, the distance from an unclassified sample to each of the training samples is computed, and the unclassified sample is assigned to the class of its nearest spectral neighbor. Cover (4) has shown that the error probability of the nearest neighbor classification is always less than twice the error probability of maximum liklihood classification using the correct probability distribution. The computation requirement for many training samples is high, but the number of distance calculations can be reduced. The training samples can be condensed to those near the boundaries of the class regions, which are sufficient to define the class discrimination (12). The training samples can be ordered by magnitude in one spectral band, so that training samples far from the unknown sample in that band don't require a distance computation (8).

If the parametric assumption is the source of classification error, nonparametric methods will improve accuracy. However, when clustering is used to divide the classes into homogeneous and separable subclasses for normal maximum liklihood classification, the resulting class distributions can be very similar to the empirical distributions. The interactive process is "frequently subjective" (19 p733) and determined by "artistic liscense" (19 p732). This requires a knowledgeable and talented system user. The use of nonparametric classifiers may not improve classification results, but can shift effort from the user to the computer, and allow less experienced users to obtain good results.

SPATIAL-SPECTRAL CLASSIFICATION

The classifiers described above operate independantly on each sample, using only spectral characteristics. Human image interpreters make use of context, edges, texture, and grey scale, in that order of importance (17 pp380, 404). Humans readily abstract spatial relations, but find it difficult to discriminate using grey levels in many bands; the opposite is true of computer algorithms.

The identification of spatial patterns is the objective of character analysis, of computer scene analysis, and of some areas of remote sensing, such as detection of roads or geologic features. Texture analysis of multispectral scanner data has been implemented by transforming spatial data to the frequency domain (17 p404), and by deriving texture measures from simultaneous higher resolution data (Bryant JPL). A simple method for using spatial context is to increase the probability estimates for the classes occuring in spatial neighbors. Spatial-spectral clustering assigns samples to the same class only if they are both spatially connected and spectrally similar (19 p781, 17 p401). Pearson of ERL and others select samples for preliminary clustering only if they are part of spectrally similar spatial blocks.

The best known method of spatial-spectral classification is per-field classification (19 p785, 17 p371), in which samples in one agricultural field are all given the same classification. The fields are defined by a boundary drawing algorithm or by spatial-spectral clustering. The per-field technique gives improved classification accuracy of crop type. Fu found 92.6% accuracy versus 79.7% accuracy for the normal maximum liklihood classifier (9 p10). The classified images have much less random classification noise than images produced ignoring spatial considerations, and appear more like the product of human drafting. The total computation requirement of per-field classification is reduced, because each field rather than each pixel is classified (15).

ERROR ESTIMATION

The quality of a classifier is determined by its error performance. There are two common methods of error estimation, resubstitution and cross-validation (20, 9). In resubstitution, the original training samples are classified, and the resultant error is taken as an estimate of the operational classifier error. A low

resubstitution error shows that no gross mistakes have been made in the classifier design, but the resubstitution error is usually optimistic. This error estimate does not indicate how well a classifier generalizes from the training samples. In cross-validation methods, different subsets of the training samples are held out of the classifier design, and the classifier is tested on these unused subsets.

More training samples provide both better classifier quality, and a better estimate of quality. However, the classifier error can be estimated even when no ground truth is available (13). Training samples are identified by clustering, spectral signature, or multidate imagery. The classifier error can be estimated from the distributions of the classified image samples. A simpler error estimate accepts the majority classification of each agricultural field as correct, and counts the pixels with other classification as errors.

LANDSAT 3

The recently launched Landsat 3 has a new band in the thermal far infrared region. Its resolution is 240 meters, rather than 80 meters as in the other bands. A thermal band has been found useful for vegetation classification, crop stress (17 p337, Millard ARC), and urban land use classification during the winter months (2). Classifiers can use the thermal band as a fifth dimension in computing clustering and classification distance parameters. Since the thermal band is well separated in spectrum from the other bands, and since it measures heat emmission rather than light reflectance, it provides increased separability of classes. Processing five band data will increase computer time, and perhaps some attention should be given to the feature selection problem: the selection, combination, and relative weighting of the different bands (features) for optimum class separability.

The second sensor on Landsat is the return beam videcon (RBV). In Landsat 1 and 2, the RBV consists of three television cameras imaging the same area as the multispectral scanner, in three spectral bands. The RBV provides greater geographic fidelity, but much less radiometric accuracy than the MSS. Because of tape recorder problems, relatively little RBV data has been acquired. In Landsat 3, the RBV consists of two television cameras, each providing a monochrome image of one-half the MSS field of view. The RBV resolution has been increased to about 40 meters.

The RBV data should be the primary source for applications which emphasize spatial information, and the MSS data should be used for spectral classification. Combining data from both sources requires mutual registration. The combined data could have the larger number of pixels of the RBV data, and six measured values per pixel - the five MSS bands and the RBV panchromatic value. Another approach would be to use the RBV data as sub-pixel texture information added to each MSS pixel. The combined data could be useful for spatial-spectral classification.

CONCLUSION

The normal distribution maximum liklihood classifier is standard in Landsat data analysis, but nonparametric or spatial-spectral classifiers sometimes provide useful improvements in accuracy. Because classifiers require large amounts of computer time, faster implementations are important. Extensive research in these areas is reported in the pattern recognition and image processing literature, but the techniques are not available in commercial software systems. It would be useful to review the methods described, and to evaluate the more promissing experimentally.

There are three components in an interactive image system; computer processing, human skill, and the man-machine interface. An advance in any component would be very valuable, but wide scale, low cost useage of Landsat data depends on transferring more of the burden to computer processing. It may be possible to substitute non-parametric classification for the interactive process of selecting and clustering training samples for the normal distribution maximum liklihood classifier. Spatial-spectral classification can produce less noisy classification maps, or detect spatial features, without human intervention. These techniques could increase the effectiveness and satisfaction of image system users.

Important areas for future work at NASA-ARC are:

1. development of a nonparametric classifier to reduce human interaction,
2. development of spatial algorithms to improve classification accuracy,
3. development of classifiers using the new bands of Landsat 3.

REFERENCES

1. A. K. Agrawala, Machine Recognition of Patterns, New York, IEEE Press, 1977.
2. L.L.Bielh, and L.F.Silva, "Machine Aided Multispectral Analysis Utilizing Skylab Thermal Data for Land Use Mapping", IEEE Trans. Geoscience Elect., vol. GE-14, pp 49-54, Jan. 1976
3. C.H.Chen, Statistical Pattern Recognition,Rochelle Park, N.J., Hayden, 1973.
4. T.E.Cover, and P.E.Hart, "Nearest Neighbor Pattern Classification", IEEE Trans. Information Theory, vol.IT-13, pp21-27, Jan. 1967.
5. R.O.Duda, and P.E.Hart, Pattern Classification and Scene Analysis,New York, Wiley, 1973.
6. ESL Inc., IDIMS User's Guide; Earth Resources Processing Function, 1977.
7. W.Eppler, "Canonical Analysis for Increased Classification Speed and Channel Selection", IEEE Trans. Geoscience Electronics,vol. GE-14, pp. 26-33, Jan. 1976.
8. J.H.Friedman, F.Baskett, and L.J.Shustek, "An Algorithm for Finding Nearest Neighbors", IEEE Trans. Comput., vol. C-24, pp1000-1006, Oct. 1975.
9. K.-S.Fu,"Pattern Recognition in Remote Sensing of the Earth's Resources", IEEE Trans. Geoscience Electronics, vol. GE-14, pp. 10-18, Jan. 1976.
10. K.S.Fu, P.J.Min, and T.J.Li, "Feature Selection in Pattern Recognition", IEEE Trans. Systems Science and Cybernetics, vol. SSC-6, pp. 33-39, Jan. 1970.
11. R.M.Haralick, "The Table Look-Up Rule", 1977 Conference on Pattern Recognition and Image Processing, IEEE Computer Society, pp. 447-454.
12. P.E.Hart, "The Condensed Nearest Neighbor Rule", IEEE Trans. Information Theory, vol. IT-14, pp. 515-516, May 1968.
13. K.A.Havens, T.C.Minter, and S.G.Thadani,"Estimation of the Probability of Error Without Ground Truth and Known A Priori Probabilities", IEEE Trans. Geoscience Electronics, vol. GE-15, pp. 147-152, July 1977.
14. Y.C.Ho and A.K.Agrawala, "On Pattern Classification Algorithms; Introduction and Survey", Proc. IEEE, vol. 56, pp.2101-2114, Dec. 1968.
15. R.L.Kettig and D.A.Landgrebe,"Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects", IEEE Trans. Geoscience Electronics,vol. GE-14, pp. 19-25, Jan. 1976.

16.D.Kroeck, Everyone's Space Handbook; A Photo Imagery Source Manual, Arcata, CA, Pilot Rock, 1976.

17.J.Lintz and D.S.Simonette, eds, Remote Sensing of Environment, Reading, Mass., Addison Wesley, 1976.

18.G.Nagy, "Digital Image-Processing Activities in Remote Sensing for Earth Resources", Proc. IEEE,vol. 60. pp. 1177-1200, Oct. 1972.

19.R.G.Reeves, A.Anson, and D.Landau, eds, Manual of Remote Sensing, Falls Church, VA, Am. Soc. Photogrammetry, 1975.

20.G.T.Toussaint, "Bibliography on Estimation of Misclassification", IEEE Trans. Information Theory, vol. IT-20, pp. 472-479, July 1974.

21.H.L.Van Trees, Detection, Estimation, and Modulation Theory, New York, Wiley, 1968.

SECTION D

SATELLITE COMMUNICATIONS

FOUR DIMENSIONAL

BANDWIDTH EFFICIENT MODULATION

CONTENTS

I. INTRODUCTION

The filling of the allocated spectrum and the use of higher power satellites have led to consideration of bandwidth efficient modulation techniques for satellite communications. Phase shift keying (PSK) has been replaced by quadrature phase shift keying (QPSK) in some applications, and multiple phase shift keying (MPSK) and two and four dimensional amplitude-phase shift keying (APSK) are being investigated. [1,2,3,4] Although MPSK and APSK have not been implemented for satellite communication, they have been used in telephone modems for several years. [5] This paper examines the theoretical basis of higher dimensional modulation, and describes two new classes of four dimensional designs that are more efficient than familiar APSK modulation techniques.

Shannon's well-known capacity bound defines the trade-off in bandwidth efficient modulation, and limits the potential gain of higher dimensional signal design. [6] Shannon proved that the largest possible number of error free messages, M, that can be transmitted over a communications channel is

$$M = \left(\frac{P+N}{N} \right)^{TW} \tag{1}$$

P is the signal power, T is the signal duration, W is the signal bandwidth, and N is the noise power in the signal bandwidth, W. M messages can be transmitted without error if the product TW is allowed to become infinitely large, and the actual probability of error decreases exponentially as TW increases. [7] If the channel capacity, C, is defined as the maximum rate, in bits per second, that error free information can be transmitted over a channel,

$$C = (1/T) \log_2 M = W \log_2 \left(\frac{P+N}{N}\right) \tag{2}$$

The channel capacity can be doubled either by doubling the bandwidth, W, or

by increasing the signal power, P, so that the signal-to-noise ratio (P+N)/N is squared.

The dimensionality, D, of the message set is the number of basis functions required to represent the message set. The two best known kinds of basis functions are the Fourier series sinusoids in the frequency domain, and the sampling (sin x)/x pulses in the time domain. The number of basis functions required to represent a message set, the dimensionality, is equal to about 2TW. [6, 8] A continuous three minute telephone message has 10^6 dimensions, and a one hour television transmission has 10^{10} dimensions. Using large signal set dimension requires long delays and high information storage. Because of its simple implementation, the most common digital communications signal is the PSK pulse, with duration, T, approximately equal to 1/2W, so that its dimensionality is one.

The purpose of bandwidth efficient modulation is to increase the transmission rate without increasing channel bandwidth, by using more signal power. To emphasize power, we express the Shannon bound as capacity per dimension, C_D. [9] The noise power spectral density is N_0, where $N_0 = N/W$, and the signal energy is $E = PT$.

$$C_D = (1/2TW) \log_2 M = C/2W$$

$$= (1/2) \log_2 \left(\frac{N+P}{N}\right) = (1/2) \log_2 \frac{WN_0 + E/T}{WN_0}$$

$$= (1/2) \log_2 (1 + 2 E_D / N_0) \qquad (3)$$

The energy per dimension is $E_D = E/D = E/2TW$.

II. CONVENTIONAL MODULATION DESIGNS

Commercial telephone modems have transmission rates from a few hund to 1,200, 2,400, or 4,800 bits per second (bps). [10] For simplicity, the modems are designed in few dimensions. The baud rate of a transmission system is the number of time division pulses per second, and is usually considerably less than twice the highest useable baseband frequency. For the voice grade telephone channel, the baud rate is 2,400, while 2W is 6,600. Modems use signals designed for a single baud, containing only one or two dimensions.

The types of modem signal designs discussed by Davey [10] are described in Table I. Two one dimensional designs, PSK and four level (L=4) amplitude modulation (AM) are added for comparison. The lowest rate system is binary frequency shift keying (FSK), which has a rate of one-half bit per dimension. One of two orthogonal frequencies is transmitted in each baud. Since the frequencies are not transmitted independantly, the design is two dimensional. The receiver detects each frequency independantly, then selects the most likely. Table I shows the signal geometry for each modulation method, indicates the design dimensionality, and gives the transmission rate per baud and per dimension. At 2,400 bps, four phase QPSK is standard. Since only one of the four quadrature signals is transmitted, the dimensions are not independent, and the design is two dimensional. The receiver detects each phase independantly, and uses both results to decide which phase was transmitted.

In addition to QPSK, there are several types of modulation used at 4,800 bps or proposed for higher rates. The two level (L=2) APSK (sometimes implemented as vestigial side band)is a one dimensional system, although it is geometrically equivalent to QPSK except for a 45 degree rotation. Instead of one of two possible orthogonal signals being transmitted, as in QPSK, both of the orthogonal signals are transmitted, and each is received and decoded independantly. All of the multi-level APSK designs are one dimensional, as the encoding and decoding of each quadrature phase is independant. APSK designs are used because their one dimensional structure permits easy implementation of high speed receivers. [2] The eight phase MPSK systems are two dimensional. It is evident from the signal geometry that the higher rate designs require more signal power for equal error probability.

Table I. Modem Signal Designs

Design Name	Signal Design	Bits per Baud	Design Dimension	Bits per Dimension
PSK		1	1	1
L=4 AM		2	1	2
Binary FSK		1	2	1/2
QPSK		2	2	1
L= 2 APSK		2	1	1
L=3 APSK		3.16	1	1.58
MPSK, 8 phase		3	2	1.5
L= 2, 4 phase		3	1	1.5
L=4 APSK		4	1	2
L=2, 8 phase		4	2	2
L=6 APSK		5.18	1	2.59
L= 8 APSK		6	1	3

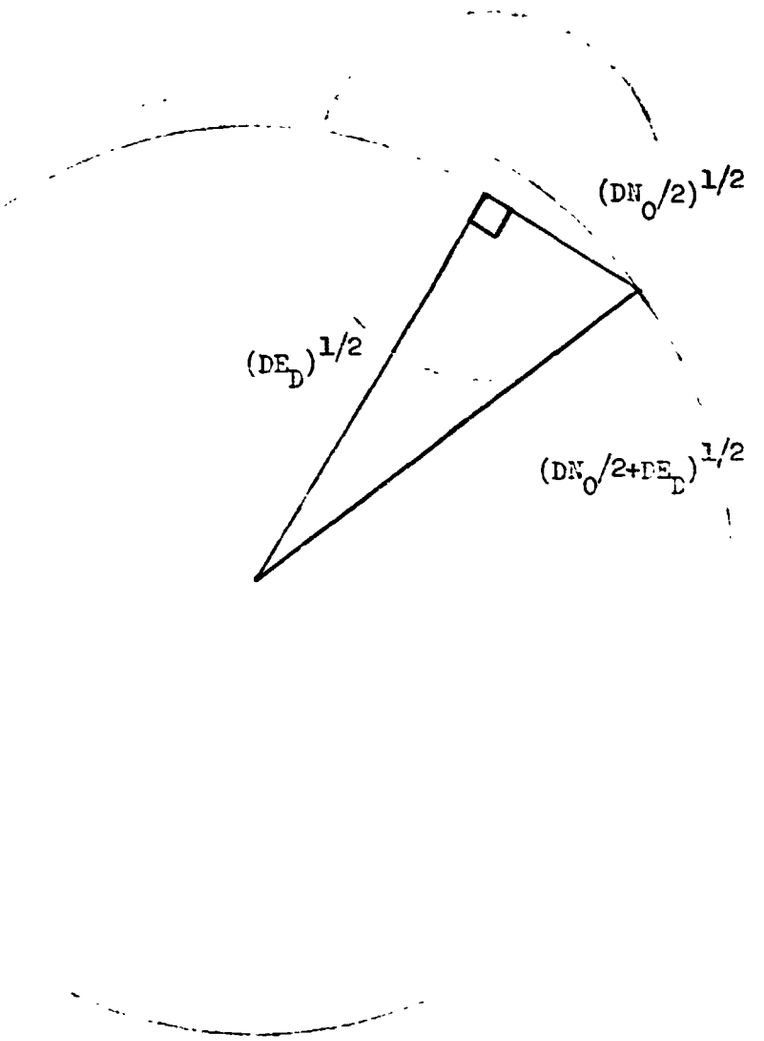
III. THE GEOMETRIC PROOF OF THE CAPACITY THEOREM

The capacity in bits per dimension is bounded by equation (3). The geometric derivation of this equation gives insight into the efficiency of modulation signal designs.[6, 11] The derivation assumes that there are M signals, each with energy less than or equal to $E = D E_D$. The transmission channel adds gaussian noise, with noise energy per dimension equal to $N_0/2$ ($DN_0/2 = 2TwN_0/2 = NT$). When D becomes large, the average noise power becomes very close to N, and the noise perturbs the signal to some point near the surface of a sphere of radius $(DN_0/2)^{1/2}$, centered on the original signal. For low error probability, the decision region of each signal must include the noise sphere surrounding the signal. The number of signals is bounded because the signals are within a sphere of radius $(DE_D)^{1/2}$, as the signal energy is bounded. Since the volume of a D dimensional sphere of radius r is $B_D r^D$, and since $B_D (r-e)^D = B_D r^D(1-r/e)^D$ is much less than $B_D r^D$, for D large, nearly all the volume of the sphere of allowed signals, and nearly all the signals, are near the surface of the sphere of radius $(DE_D)^{1/2}$. Although the length of the noise perturbation is very nearly equal to $(DN_0/2)^{1/2}$, the noise distribution along the direction of the signal is gaussian, of zero expected value, and the noise perturbation is nearly orthogonal to the signal. The signal vector and an orthogonal noise vector are shown in Figure 1. The figure allows the computation of the number of messages that can be distinguished after transmission. The volume that must be allowed for each signal is $B_D (DN_0/2)^{D/2}$, the volume of the noise sphere. The total volume available for the noise spheres is $B_D (DN_0/2 + DE_D)^{D/2}$. The bound on M is

$$M \leq \left(\frac{DN_0/2 + DE_D}{DN_0/2} \right)^{D/2}$$

$$C_D = (1/D) \log_2 M = (1/2) \log_2 (1 + 2 E_D / N_0) \qquad (4)$$

This is identical to equation (3).



Figurel. The geometric proof of the upper bound on capacity.

IV. THE PERFORMANCE OF ONE DIMENSIONAL APSK

The above geometric derivation of the capacity theorem, with the converse proof that the bound can be approached for large D, implies that in efficient signal designs, each signal decision region includes the noise sphere without much additional volume, and that all the decision regions fill the region of allowed signals. This result for large dimension is meaningful for one dimensional APSK, because APSK is actually used in a large number of dimensions. Consider the simplest case of two level APSK or PSK. During each pulse period, a value of ±1 is transmitted. As the transmission is repeated, in sucessive pulse periods or dimensions, the possible signal points form a one dimensional array, then a two dimensional square, a three dimensional cube, and so on. After D pulse periods, the transmitted signal is a D dimensional vector $(\pm 1, \pm 1, \pm 1, \ldots \pm 1)$, which is one of the 2^D vertices of a D dimensional hypercube. Although it was encoded, transmitted, and decoded as a sequence of one dimensional signals, the same vector could be the result of a D dimendional design, for example a length D block code. Block coding reduces the number of possible signals below 2^D by using parity check bits, and so improves error performance by reducing the rate per dimension. In contrast to bandwidth efficient modulation, lower signal to noise is used while bandwidth is increased.

How efficient is one dimensional APSK as D becomes large? We first observe, after Wozencraft and Jacobs [12], that APSK leaves empty some of the volume allowed for the received signal plus noise. In Figure 2, we see that there is unfilled area near $\pm(DE_D)^{1/2}$ on the orthogonal axes. We also observe that the decision regions for each signal are cubic, rather than the optimum spherical shape. We compute the number of possible messages for APSK, and compare it with the capacity bound for large D.

The maximum signal amplitude in D dimensions is $(DE_D)^{1/2}$. This is the length from the center of the APSK cube to the most distant signal. If each edge of a D dimensional cube has length 1, the cube diagonal has length $D^{1/2}$. The length of the diagonal is twice the length of the largest signal, $2(DE_D)^{1/2}$, so that each edge of the APSK cube has length $2E_D^{1/2}$, and extends from $-E_D^{1/2}$ to $+E_D^{1/2}$ in each dimension. The transmitted signals are all in or on a cube of

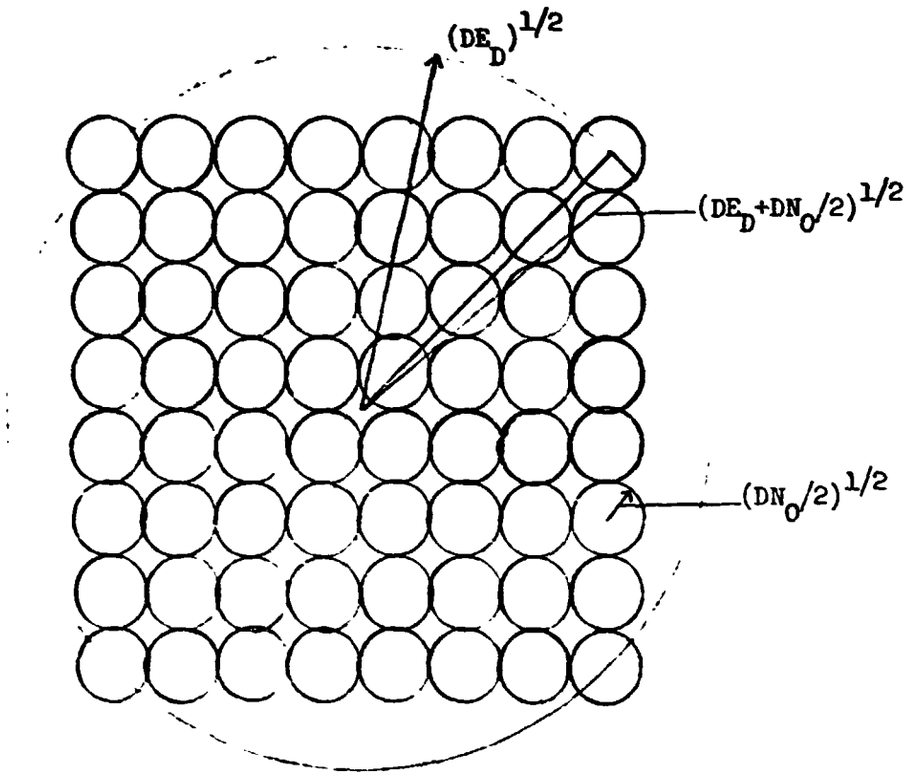


Figure 2. Two dimensional section of APSK.

side $2E_D^{1/2}$. As D becomes large, the decision region for each signal must include the noise sphere of radius $(DN_0/2)^{1/2}$, as in the capacity theorem proof. The decision regions for APSK are D dimensional hypercubes. To contain a sphere of radius $(DN_0/2)^{1/2}$, they must have an edge of $2\ (DN_0/2)^{1/2}$. The received signal decision regions are then all contained in a hypercube of edge $2\ (DN_0/2)^{1/2} + 2\ E_D^{1/2}$.

The number of APSK signals is exactly equal to the volume of the hypercube containing all the decision regions divided by the volume of one hypercube decision region.

$$M_{APSK} = \frac{(\ 2(DN_0/2)^{1/2} + 2\ (\ E_D)^{1/2}\ )^{D}}{(\ 2\ (DN_0/2)^{1/2}\ )^{D}}$$

$$= (\ 1 + (2\ E_D/\ DN_0)^{1/2}\ )^{D}$$

The rate per dimension, R_D , is

$$R_{D,\ APSK} = (1/D)\ \log_2 M = \log_2\ (1 + (2\ E_D/\ DN_0)^{1/2}\ ) \tag{5}$$

For simplicity in comparison , we assume that E_D is much greater than $DN_0/2$, which implies a nigh rate per dimension. Under this assumption, the following equations are approximately correct.

$$R_{D,\ APSK} = (1/2)\ \log_2\ (2\ E_D/DN_0)$$

$$C_D = (1/2)\ \log_2\ (2\ E_D\ /N_0)$$

$$R_{D,\ APSK} = (1/2) \log_2 D \qquad (6)$$

Each time D is multiplied by 4, the achievable rate drops one bit per dimension below capacity. For $D = 10^6$, approximately 4^{10}, which is the dimensionality of a three minute telephone message, the rate of APSK is reduced 10 bits per dimension below capacity. Although this result holds only for large dimension and large signal to noise, it is possible that four dimensional designs might gain up to one bit per dimension over one dimensional APSK.

One dimensional APSK is a poor signal design for large dimension. This is because a cube is not very similar to a sphere in higher dimensions. The two causes of the inefficiency of APSK in large dimension, are that the cube of signal points occupies only a small portion of the circumscribed sphere of allowed signals, and that the noise spheres required for low error occupy a small portion of the circumscribed cubic decision regions. Good signal designs require a high density packing of spherical decision regions that extends throughout the sphere of allowed signals.

V. TWO DIMENSIONAL SIGNAL DESIGN

APSK is the best one dimensional signal design, since the sphere of allowed signals and the spherical decision regions reduce to line segments in one dimension. The optimum design speces signals equally along a line. Two dimensional signal designs offer some scope for improvement. They have been studied by Foschini, Gitlin, and Weinstein [13]. It is well known that the densest packing of non-intersecting spheres (the decision regions) in two dimensional space is defined by placing the sphere centers on points of the equilateral triangle or regular hexagon lattice, shown in Figure 3. For large M, the optimum signal design is a circular region of the equilateral triangle lattice. For small M, the locally optimum signal designs are circular regions containing slightly irregular approximations to the equilateral triangle lattice. Foschini and his co-authors found that the best designs for 8 and 16 signals are only 1.0 dB and 0.5 dB better than one dimensional APSK.

The potential of two design can be easily bounded. The gain in useable signal volume from using all the region with less than the maximum signal energy is the ratio of the area of a circumscribed circle to the area of an inscribed square. If r is the radius of the circle, the ratio is

$$\text{ratio, full area} = \frac{\pi r^2}{2\ r^2}$$

The gain in number of signals per unit volume from the use of the equilateral triangle lattice is the ratio of the APSK decision volume to the equilateral lattice decision volume. In a lattice structure, there is one signal for each lattice cell, and each signal has a decision volume equal to the volume of one lattice cell. While the decision regions in a square lattice are squares, the decision regions in an equilateral triangle are hexagons, as shown in figure 3, and the lattice cells are parallelopipeds. The area of a square cell of edge 2 is 2^2 or 4. The area of the parallelepiped cell of edge 2 in the equilateral triangle lattice is $2\ 3^{1/2}$. The ratio of gain in signals per unit volume is

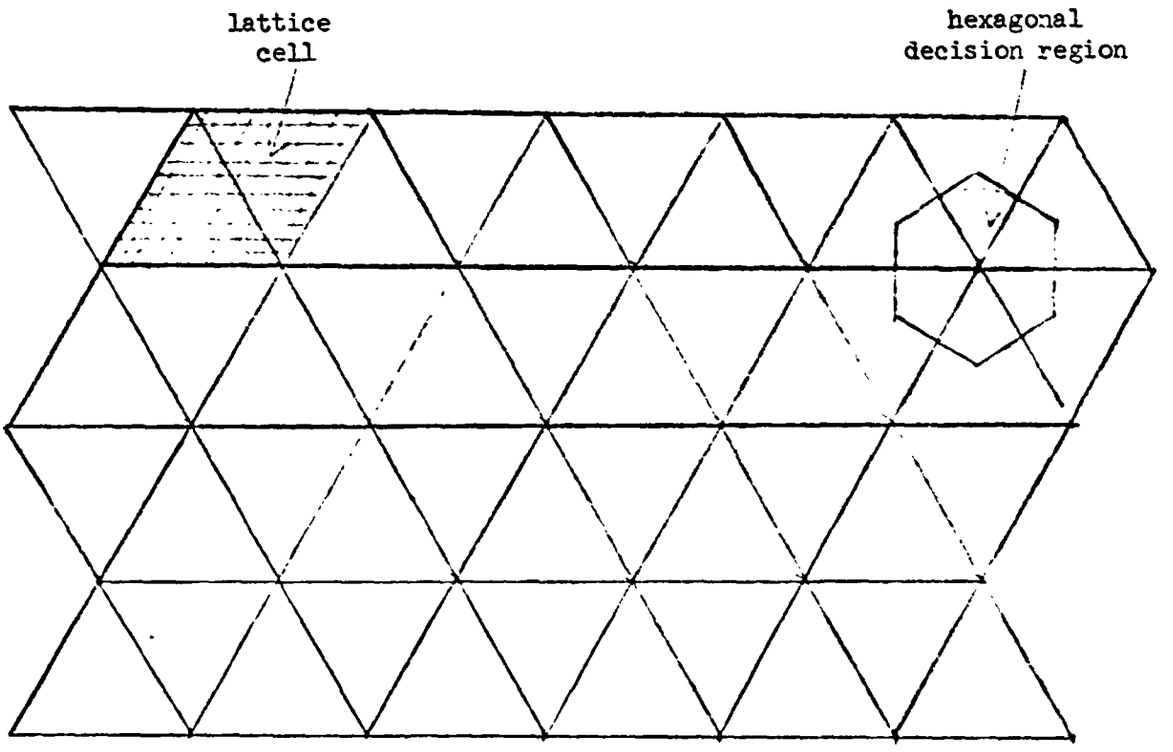


Figure 3. Equilateral triangle-regular hexagon lattice.

$$\text{ratio, triangle lattice} = \frac{4}{2\,3^{1/2}} = \frac{2}{3^{1/2}}$$

Considering both the increased area for signals, and the reduced area required for each signal, the number of signals in one dimensional APSK can be increased by the product of the above ratios, for two dimensional design.

$$\text{ratio} = \frac{\pi}{2} \quad \frac{2}{3^{1/2}} = \frac{\pi}{3^{1/2}}$$

The maximum two dimensional design rate per dimension is

$$R_{MAX} = (1/2) \log_2 (M_{APSK} * \text{ratio})$$

$$= R_{APSK} + (1/2) \log_2 \frac{\pi}{3^{1/2}}$$

$$= R_{APSK} + 0.576$$

If the number of signals is large, as for large signal to noise, so that little area is wasted, two dimensional design can approach a rate 0.576 bits per dimension higher than APSK.

Suppose that, instead of increasing the number of signals transmitted with a given energy, it is desired to keep the same number of signals and reduce the maximum signal energy. Since the number of signals in a fixed area is increased by $\pi / 3^{1/2}$, a fixed number of signals can be contained in an area reduced by $3^{1/2}/\pi$. Since the area is $\pi r^2 = \pi (E^{1/2})^2 = \pi E$, the energy E can be reduced by $3^{1/2} / \pi$ or 2.60 dB. The APSK and equilateral lattice signal separation are the same, but the number of neighboring signals is increased from 4 to 6, and the probability of error probably increases 50 percent.

VI. FOUR DIMENSIONAL SIGNAL DESIGN

We will consider signal designs in four dimensions, which are expected to provide more improvement than two dimensional design. Four dimensional designs have been suggested for satellite communications by Welti and Lee [14] and Welti [15]. The four orthogonal signals defining the four dimensions can be two sucessive pulses, each with two phase-quadrature signals, but any four orthogonal signals give equivalent performance. Comparison of APSK to the capacity bound, equation (6) above, indicated that it may be possible to increase the transmission rate by one bit per dimension using four dimensional design, without increasing signal energy and while maintaining the same minimum signal separation and approximately the same error probability.

We first define the gain achieved by filling the entire allowed region with signals, rather than using only the hypercube APSK region. Suppose that a four dimensional cube has edge equal to X. The cube diagonal is $(4\ X^2)^{1/2}$, and since this is equal to $2\ E^{1/2}$, the maximum signal amplitude is also X. The hypercube volume is X^4, and the volume of the circumscribed hypersphere is $(1/2)\ \pi^2\ X^4$, after Sommerville [16]. The ratio of the allowed signal volume to the volume used by APSK is

$$\frac{(1/2)\ \pi^2\ X^4}{X^4} = \frac{\pi^2}{2}$$

Assuming that the number of signals increases proportionately, the increased rate per dimension is $(1/4)\ \log_2\ (\pi^2/2) = 0.58$ bits per dimension. The volume of the allowed signal sphere is $(1/2)\ \pi^2\ (E^{1/2})^2 = (1/2)\ \pi^2\ E^2$. If the same number of signals is reduced, the volume can be reduced by $2/\pi^2$, and the signal energy can be reduced by $2^{1/2}/\pi$, or 3.46 dB.

This gain can be obtained by extending the four dimensional hypercube lattice of APSK throughout the hypersphere defined by the maximum signal energy. If the signal separation is 2, the radius of the sphere included in the decision region is 1. Then the APSK design can have a signal placed at the point (0,0,0,0), and

additional signals placed at the points defined by the four orthogonal displacement vectors

(2,0,0,0)
(0,2,0,0)
(0,0,2,0)
(0,0,0,2)

The minimum signal separation is 2, and additional signal points can be added by adding or subtracting these four displacement vectors from any previous signal point. All lattice points $(\pm 2a, \pm 2b, \pm 2c, \pm 2d)$ such that $2(a^2+b^2+c^2+d^2)$ 1/2 is less than or equal to $(4E_D)^{1/2} = E^{1/2}$, do not exceed the allowed signal energy.

Suppose that a point at (1,1,1,1) is added to the four dimensional hypercube lattice. This point, like the original members of the hypercube lattice, has a distance of 2 from all the hypercube lattice points, and therefore has a decision region which includes a sphere of radius 1. (Distance is computed by the Euclidean formula, $d = (w^2+x^2+y^2+z^2)^{1/2}$, where w,x,y, and z are distances on orthogonal axes.) For each point in the original hypercube lattice, a new point can be added at displacement (1,1,1,1). The number of messages in the allowed signal hypersphere is doubled. It is obvious that the transmission rate is increased by one bit in four dimensions, or 0.25 bits per dimension.

The volume of each decision region in the original hypercube lattice is 2^4, but the decision volume is reduced to 2^3, and is no longer cubic, when the number of signals is doubled. Since the volume per decision region is one-half the original volume, the original number of signals can be accomodated in one-half the original volume, in the new lattice. The hypersphere volume is $(1/2)\ \pi^2 E^2$, so that E can be reduced $1/2^{1/2}$, or 1.51 dB.

The new denser lattice can be described as a body centered cubic lattice, to indicate the signal added at the cube center. The cube center signal has sixteen neighbors at the vertices of the original hypercube, defined by (0,0,0,0), (2,0,0,0), (0,2,0,0,), ... , (2,2,2,2), and eight neighbors at the centers of eight neighboring

hypercubes, located at (1,1,1,1) displaced by (±2,0,0,0), (0,±2,0,0), (0,0,±2,0), and (0,0,0,±2). Each hypercube vertex signal similarly has twenty-four neighbors. In addition to the eight original lattice signals at (±2,0,0,0), (0,±2,0,0), (0,0,0,±2), and (0,0,0,±2), it has new neighbors at the centers of the sixteen hypercubes that meet at (0,0,0,0). These signals are at the vertices of a hypercube (±1,±1,±1,±1).

An identical lattice can also be constructed as an alternate vertex cubic lattice. This construction will also be given, since it uses the familiar idea of a parity check, and provides a simpler description of signal designs and the required receiver. Consider the cubic lattice defined by the zero point (0,0,0,0) and the four orthogonal vectors

$$(\pm 2^{1/2},0,0,0)$$
$$(0,\pm 2^{1/2},0,0)$$
$$(0,0,\pm 2^{1/2},0)$$
$$(0,0,0,\pm 2^{1/2})$$

If these defining vectors are designated as $\bar{X}_1$, $\bar{X}_2$, $\bar{X}_3$, and $\bar{X}_4$, the cubic lattice points are all at the tips of vectors of the form $a_1 \bar{X}_1 + a_2 \bar{X}_2 + a_3 \bar{X}_3 + a_4 \bar{X}_4$, where a_1, a_2, a_3, and a_4 are integers. If a parity check is placed on the sum of the vector coordinates, so that $a_1 + a_2 + a_3 + a_4$ must be even or odd, the vector tips are points on an alternate vertex cubic lattice. The signal separation of the original cubic lattice is $2^{1/2}$. This separation is increased to 2 by the parity requirement, which insures that if two signals differ by $2^{1/2}$ in one dimension, they differ by $2^{1/2}$ in two orthogonal dimensions. The alternate vertex cubic lattice has the same signal separation as the body centered cubic lattice. The volume of the original cubic decision region is $(2^{1/2})^4 = 4$. This volume is doubled, and changed in shape, when only the even or odd parity vertices are used. The alternate vertex cubic lattice has the same decision volume as the body centered cubic lattice.

It can be shown that each signal in the alternate vertex cubic lattice has 24 neighbors, as in the body centered cubic lattice. For the alternate vertex cubic lattice, the number of neighbors is the number of ways to add two orthogonal vectors of value $\pm 2^{1/2}$ to the given signal. There are six ways to choose 2 out of four dimensions, and 4 ways to assign two signs, so each signal has 24 neighbors. If the four dimensions are w,x,y, and z, the 6 pairs are wx, wy, wz, xy, xz, and yz. Each pair defines a two dimensional plane section of the alternate vertex cubic lattice, as shown in figure 4. A signal has 4 neighbors in each plane. In both constructions, each neighboring signal has 4 neighbors which are at distance 2 from itself, and also at distance 2 form the original signal.

Although the body centered cubic lattice and the alternate vertex cubic lattice are identical structures, the original defining lattices differ in scale and rotation. The original lattice for the body centered cubic construction has one-half the final number of signals, and the original lattice for the alternate vertex cubic construction has twice the final number.

It is not difficult to visualize four dimensional structures. Figure 5 is a two dimensional hypercube, similar to the familiar projections of the three dimensional cube. The projection contains 8 interlocking three dimensional cube projections. The hypercube projection contains 16 vertices, where 4 lines meet. These 4 lines meeting at each vertex are the 4 orthogonal vectors, $\pm w$, $\pm x$, $\pm y$, and $\pm z$. Each dimension is always represented by the same orientation in the projection, as shown. To go from point (0,0,0,0) to (1,1,1,1), it is necessary to move along a vector from o to 1 in each dimension. Each restrictive equation, say w = 0, reduces the dimensionality of the figure by one. The 8 three dimensional cube projections are defined by w = 0,1, x = 0,1, y = 0,1 and z = 0,1. The 8 three dimensional cube projections are joined in pairs. When one three dimensional cube is identified, it can be seen that there is an extra line having the same slope at each vertex. Following all these parallel lines to the next vertex identifies a new three dimensional cube, parallel to the original in four dimensional space. The bounding of the four dimensional cube by three dimensional cubes is similar to the bounding of three dimensional cubes by squares, and the bounding of squares by line segments. If a body centered point is added, it is located in the center of the projection. The vertices of even parity are circled.

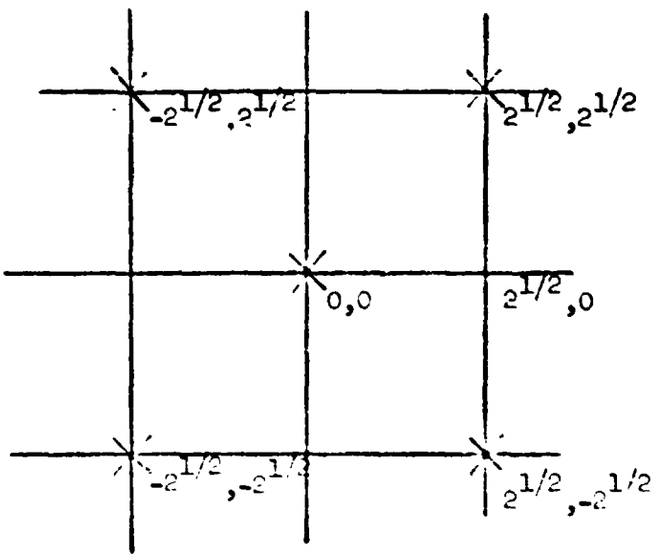


Figure 4. Two dimensional section of the alternate vertex cubic lattice.

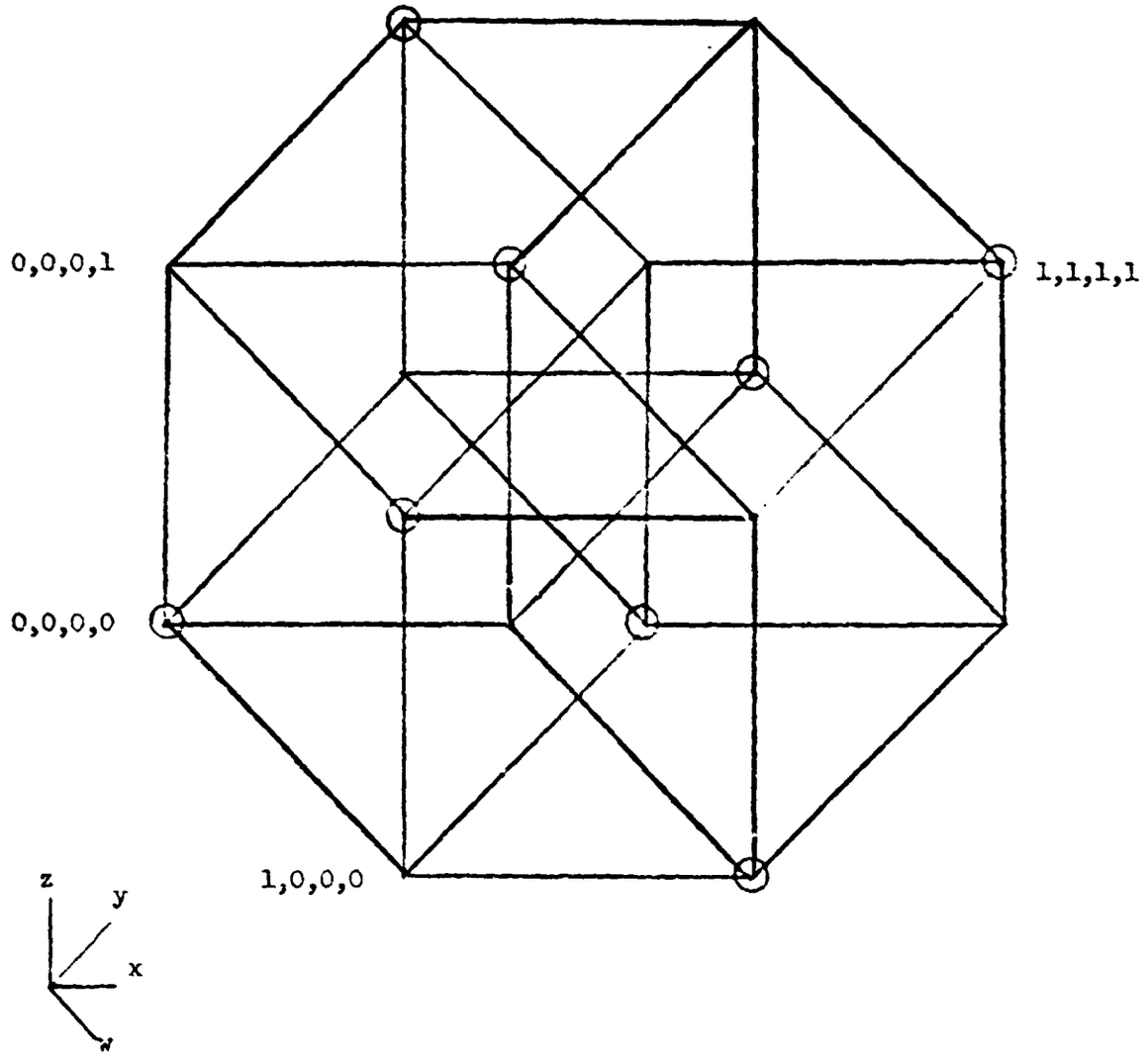


Figure 5. A two dimensional projection of the four dimensional hypercube.

The alternate vertex cubic lattice is the densest lattice in four dimensions, as discussed by Leech [17]. There is no possible denser packing of the non-intersecting noise spheres contained in the decision regions. Since it has been shown that circular regions of the densest two dimensional lattice approximate the most efficient two dimensional signal designs, it can be reasonably conjectured that four dimensional hypersphere regions of the densest four dimensional lattice will approximate the best four dimensional designs.

Suppose that it is desired to increase the number of signals while using the same maximum signal energy. Using all the available signal volume gains 0.58 bits per dimension, and using the densest lattice gains 0.25 bits per dimension, so that using both produces a rate increase of 0.83 bit per dimension. Suppose that it is desired to keep the same number of signals and reduce maximum signal energy. Since the number of cubic lattice signals is increased by $\pi^2/2$, by using all the volume, a fixed number of signals can be contained in a volume reduced by $2/\pi^2$. Since the four dimensional volume is $(\pi^2/2)\ E^2$, E can be reduced by $2^{1/2}/\pi$, or by 3.46 dB. Similarly, using the densest lattice increases the number of signals by 2, a fixed number of signals can be contained in a volume reduced by 1/2. E can be reduced by $1/2^{1/2}$, or 1.51 dB. Using both all available volume and the densest lattice, E can be reduced by $1/\pi$, or 4.97 dB. These gains are limits that can be approached for high signal to noise, and a large number of signals. For low signal to noise, crowding may cause wasetd volume. Although the signal separation is held constant, increasing the number of neighbors from 16 to 24 in the densest lattice design probably increases error probability about 50 percent.

VII. SPECIFIC CONFIGURATIONS OF FOUR DIMENSIONAL SIGNALS.

The familiar one and two dimensional signal designs of Table I have rates between 1 and 3 bits per dimension. We will describe the rate per dimension and the ratio of maximum signal energy to signal separation for one dimensional APSK, the extended cubic lattice, and the alternate vertex cubic lattice.

Because of the simplicity of the one dimensional APSK, the required parameters are easy to compute. Suppose that the APSK design has L levels. For L odd, the L different signal levels are $(L-1)/2$, $(L-3)/2$, ..., 2, 1, 0, -1, -2, ... $-(L-1)/2$. For L=1, (0,0,0,0) is the only signal. For L=3, the value in each dimension can be -1, 0, or 1. The number of signals is $M = L^4$, in four dimensions, and the rate per dimension $R = (1/4) \log_2 M = \log_2 L$. Since the error behavior depends on the ratio of signal separation to noise variance, signal designs at a given rate can be ranked by the ratio of peak signal amplitude to signal separation. The signals are contained in a hypercube, having L-1 units on an edge. The hypercube diagonal is twice the length of the maximum energy signal.

$$(4(L-1)^2)^{1/2} = 2\,E^{1/2}$$

$$4(L-1)^2 = 4\,E$$

$$E^{1/2} = L-1$$

The distance between signals, d, is 1 unit.

$$E^{1/2}/d = L-1$$

For this case, which will be designated case 1, L is odd and (0,0,0,0) is a signal point. For case 2, L is even and the possible signals in each dimension are L-1, L-3, ..., 5, 3, 1, -1, -3, -5, ... -L+1. For L=2, there are 16 sign[illegible],

$(\pm1,\pm1,\pm1,\pm1)$, the hypercube vertices. For case 2, $M = L^4$, as before. The hypercube diagonal is twice the largest signal.

$$(4(2(L-1))^2)^{1/2} = 2\,E^{1/2}$$

$$4(L-1) = 2\,E^{1/2}$$

$$E^{1/2} = 2(L-1)$$

The minimum signal distance, d, is 2 units, so that $E^{1/2}/d$ is as before.

$$E^{1/2}/d = L-1$$

The transmission rate per dimension, and the ratio of peak signal to signal separation are the same for L odd or even. $E^{1/2}/d$ is plotted versus R in figure 6, for L = 1, 2, ... 8.

The two new classes of signal design introduced above will be described. In the first class, the cubic lattice is extended beyond the APSK hypercube, to include all cubic vertex points not exceeding the maximum allowed signal energy. In the second class of designs, the alternate vertex cubic lattice is used in the same allowed signal region. The permutation codes of Slepian [18], which are described below, are alternate vertex cubic lattice designs in which all signals have the maximum energy. In a small number of dimensions, such equal energy designs have significantly lower rate than designs including points with less than the maximum energy. Slepian did not specifically consider the alternate vertex cubic lattice or dimension of four. Welti and Lee [14], in the consideration of four dimensional designs, include some examples of APSK and of the two new general classes of designs, but do not mention the alternate vertex cubic lattice.

In the extended cubic lattice signal design, all points of the cubic lattice having less than the maximum signal energy are used as signals. For case 1, the

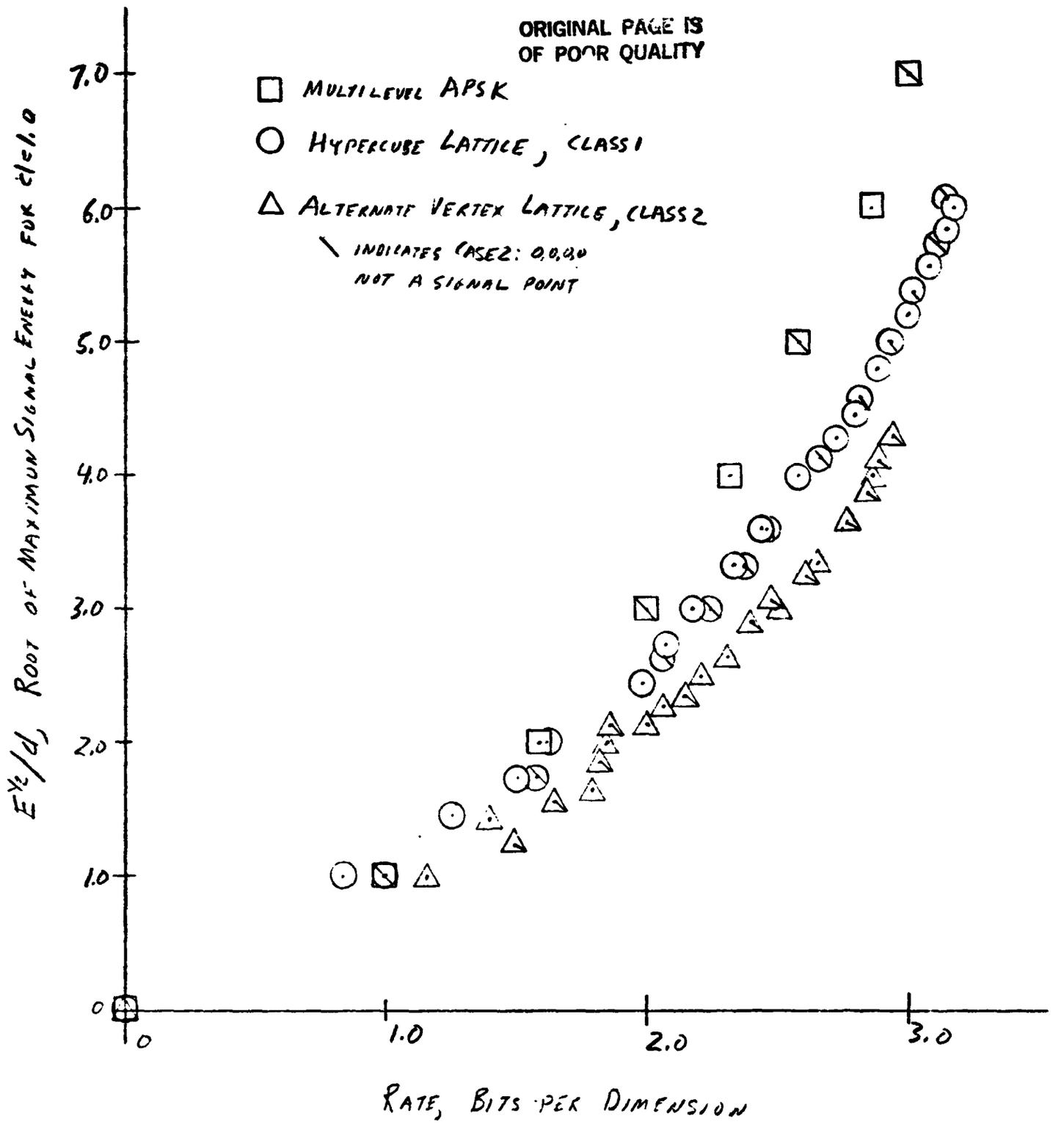


FIGURE 6. COMPARISON OF ENERGY AND RATE FOR FOUR DIMENSIONAL SIGNAL DESIGNS

cubic lattice includes all points of the form (a,b,c,d), where a,b,c, and d are any integers. The signal points can be enumerated in order of increasing energy. There is 1 point with E=0, (0,0,0,0); there are 8 points with E=1, (±1,0,0,0), (0,±1,0,0), etc.; and there are 24 points with E=2, (±1,±1,0,0), etc. Any lattice point (a,b,c,d) has energy, $E= a^2 + b^2 + c^2 + d^2$. If the value a in the first dimension is exchanged with the value b in the second dimension, or if a is replaced by -a, the energy, E, is unchanged. Thus, any lattice point (a,b,c,d) defines a shell of constant energy, containing all the points described by permutations and sign changes of the values a,b,c, and d. These equal energy signal sets were called permutaion codes by Slepian [13], who gave the formula for enumeration of the points on a shell.

The enumeration of the signals in an equal energy shell can be briefly explained. Suppose we have a lattice point in D dimensions, $(a_1, a_2, a_3,...a_D)$. We can select one of D values for the first dimension, one of D-1 values for the second dimension, etc., so that there are D! possible orders of the coefficients. If each may be positive or negative, there are $D!\ 2^D$ signals. This number must be reduced if some of the a_i have the same magnitude, and are therefore indistinguishable, or if some of the a_i are zero, so that the sign is meaningless. If the ceofficients include m_o zeros, the number of points in the shell is $D!\ 2^{D-m_0}$. If the next largest coefficient is repeated m_1 times, the $m_1!$ ways of ordering this coefficient in the m_1 dimensions selected for it are indistinguishable. The formula for the number of signals in a shell is

$$M = \frac{D!\ 2^{D-m_0}}{m_0!\ m_1!\ \cdots\ m_k!}$$

k is the number of different coefficient magnitudes.

In this work, D = 4. The number of points on the shell of (1,0,0,0) is $4!\ 2^{4-3}/3!\ 1! = 8$, corresponding to four dimensional locations for the 1, and positive or negative sign in each location. The number of points on the shell of (1,1,0,0) is $4!\ 2^{4-2}/2!\ 2! = 24$. The shells for E less than or equal to 16

are given in Table II. For several values of E, including 2 nad 3, two different sets of integers have tha same E. The total number of messages, M, for any peak signal energy, E, is obtained by adding the counts of the included shells. Since the minimum signal separation is 1, $E^{1/2}/d = E^{1/2}$. Table II was extended to E = 36, and selected designs were plotted in Figure 6. As expected, the extended hypercube lattice designs are superior to APSK, since for fixed signal separation the full latice provides higher rate at the same maximum signal energy or lower energy at the same rate.

Table II can also be used to enumerate the messages in alternate vertex lattice signal designs. The alternate vertex cubic lattice consists of the cubic lattice points having the sum of their coordinates even or odd. Each permutation shell contains points having the same parity, because permuting the coefficient order does not alter the sum of the coefficients and because changing the sign of a_i changes the coefficient sum by 2 a_i, which does not alter parity. Each permutation shell lies entirely in one of the two alternate vertex lattices, defined by even or odd parity. The shell parity is odd when E is odd, and even when E is even. This is shown as follows. If the sum of the coordinates is even, there is an even number (possibly zero) of odd coordinates. Since the squares of even numbers are even and the squares of odd numbers are odd, E is the sum of even numbers and an even mumber (or zero) odd numbers, and E is even. Similarly, if the sum of the coordinates is odd, there is an odd number of odd coordinates, and E is odd. The alternate vertex signal design includes all the shells for E even or odd. The extended ($E \geq 36$) Table II was used to compute M. Since the minimum signal distance is $2^{1/2}$, $E^{1/2}/d = (E/2)^{1/2}$. Selected alternate vertex designs are plotted in Figure 6.

A table similar to Table II was prepared for case 2, in which only vectors of all odd coordinates are signal points. Some of the points derived using this table for the full cubic lattice design and for the alternate vertex lattice design are also plotted in figure 6, and these points are specially designated. When R is small, the displacement of the center of the signal set from a vertex point to a cell center generates different combinations of $E^{1/2}/d$ and R, but the signal designs for R large are similar.

Table II. Permutation shells.

Energy, E	Defining Point	Number of Messages, M
0	(0,0,0,0)	1
1	(1,0,0,0)	8
2	(1,1,0,0)	24
3	(1,1,1,0)	32
4	(1,1,1,1)	16
	(2,0,0,0)	8
5	(2,1,0,0)	48
6	(2,1,1,0)	96
7	(2,1,1,1)	64
8	(2,2,0,0)	24
9	(2,2,1,0)	96
	(3,0,0,0)	8
10	(2,2,1,1)	96
	(3,1,0,0)	48
11	(3,1,1,0)	96
12	(2,2,2,0)	32
	(3,1,1,1)	64
13	(2,2,2,1)	64
	(3,2,0,0)	48
14	(3,2,1,0)	192
15	(3,2,1,1)	192
16	(2,2,2,2)	16
	(4,0,0,0)	8

Figure 6 shows that, compared to APSK, the two new classes of signal designs for four dimensions can be used to achieve higher rate per dimension for the same maximum signal energy and signal separation. Comparison of APSK to the capacity bound showed that rate gains of up to 1 bit per dimension might be achieved by four dimensional design. Computation showed that extending the cubic lattice throughout the allowed signal region could gain up to 0.58 bits per dimension. From Figure 6, the better designs at higher $E^{1/2}/d$ actually gain 0.35 bits per dimension. Computation also showed that use of the alternate vertex cubic lattice could gain an additional 0.25 bits per dimension. This is achieved at higher $E^{1/2}/d$, and is sometimes slightly exceeded because the alternat · vertex designs are derived from cubic designs with higher $E^{1/2}/d$, having less acidentally wasted volume. The largest total gain is about 0.6 bits per dimension, which is a substantial portion of the computed total gain of 0.83 bits per dimension.

Figure 6 also shows how much the two new classes of signal design allow the energy to be reduced at a given rate and signal separation. Previous computation showed that use of the full cubic lattice would allow an energy reduction of 3.46 dB. As shown by the figure, the full cubic lattice allows the signal energy, E, to be reduced 1.58 dB at 2 bits per dimension, and 2.58 dB at 3 bits per dimension. In the computation of the previous section, the use of the alternate vertex cubic lattice gave a further reduction of 1.51 dB, for a total of 4.97 dB. Actual alternate vertex designs allow the energy to be reduced 2.90 dB at 2 bits per dimension, and 3.95 dB at 3 bits per dimension. While the full computed gain should be achieved at high rate and signal to noise, the actual gains for useful rates are a significant portion of the possible gains.

VIII. FOUR DIMENSIONAL SIGNAL RECEIVERS

For equally probable signals, the receiver should select that member of the transmitted signal set which is closest to the received signal. This can be done, in general, by computing the distances between the received signal and all the members of the signal set, and has been suggested for four dimensional designs by Welti and Lee [14]. This complexity is obviously not required for APSK, and for the class of designs using the extended cubic lattice, since each dimension can be detected independantly. In the full cubic lattice designs, the results of four one dimensional receiver operations are used to decode the signal message. Distance computations can reduce the error probability, when errors cause the selection of lattice points beyond the allowed signal sphere.

The receiver for the alternate vertex cubic lattice is only slightly more complex than the receiver for APSK or the full cubic lattice. The receiver operation will first be outlined, and then explained in detail. The receiver initially detects the signal operation in each dimension, as if the signal set were the full cubic lattice with signal separation $2^{1/2}$. The detection process makes a primary decision, defining each orthogonal signal to some multiple of $2^{1/2}$, and also retains 3 bits of secondary information, indicating which of the four orthogonal signals has the largest error, and in what direction. If the primary decision lattice point is verified as a member of the alternate vertex cubic lattice by the parity check, it is accepted as the correct transmitted signal. If the primary decision lattice point is not a member of the alternate vertex lattice, the transmitter signal closest to the received signal is one of the primary decision point's eight neighboring alternate vertex lattice points. These lie at $\pm 2^{1/2}$ along each orthogonal dimension form the primary decision point.

To understand the decision regions and the effect of noise, refer to Figure 7. If the error in each orthogonal signal is less than $2^{1/2}/2$, the transmitted signal is correctly defined by the primary decision. The primary decision region is the decision region of a point in the full original cubic lattice. If one orthogonal signal has an error greater than $2^{1/2}/2$, and the other three signals have error less than $2^{1/2}/2$, the primary decision selects a point which is not a member of

the alternate vertex lattice. To attempt the correct decision, the original cubic lattice decision region must be augmented by portions of the original cubic decision regions belonging to deleted neignbor points. The deleted point indicated by the primary decision is equally distant from 8 neighboring alternate vertex points, lying at $\pm 2^{1/2}$ along the four orthogonal axes, and the 8 required alternate vertex decision regions intersect at the primary decision point. The needed decision information is the error vector from the primary decision point to the received signal.

The decision region of a signal includes all volume closer to that signal than any other signal. This decision rule can be implemented by (1) retaining the error vector, (2) adding it to the primary decision point, (3) computing the exact distances to the 8 neighbors of the primary decision point, and (4) selecting the signal with smallest distance to the received signal. An equivalent computation (1) takes the scalar or dot product of the error vector with the 8 displacement vectors between the primary decision point and the neighbors, and (2) selects the signal lying in the direction closest to the error vector, that is, selects the signal with the largest scalar product. Since the 8 neighbor signals lie along the four orthogonal signals defining the four dimensions, this is the equivalent of choosing the signal in the direction having the largest estimated error, according to the secondary information provided by the reciever.

We further discuss the decision regions, and the causes of a receiver error. Since the minimum signal distance is 2, The receiver always selects the correct signal when the error is less than 1, the radius of the hypersphere contained in the decision region. The primary decision is correct when the error magnitude is less than $2^{1/2}/2$ in each dimension, since this defines the full cubic lattice decision region. As can be seen from Figure 7, if the error in any dimension is greater than $2^{1/2}$, or if the error in each of two dimensions is greater than $2^{1/2}/2$, the received signal is closer to an incorrect alternate vertex point than to the correct one, and the correct decision is impossible. The receiver sometimes selects the correct signal for error magnitueds between 1 and $2^{1/2}$, since the decision regions extend $\pm 2^{1/2}$ along the orthogonal signal directions.

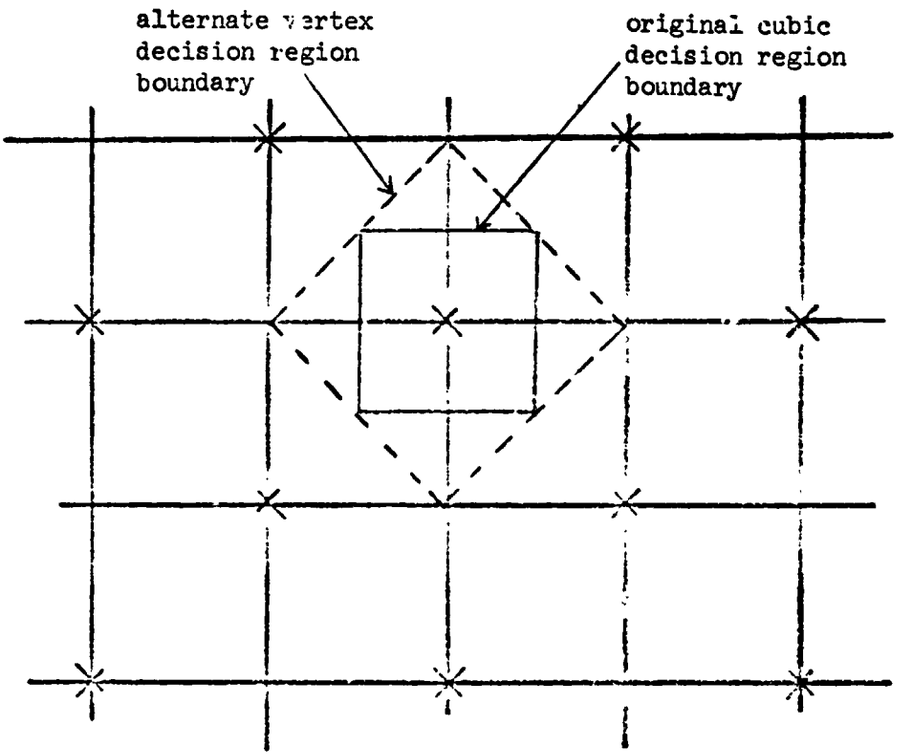


Figure 7. The cubic lattice in two dimensions, with the alternate vertices indicated by crosses, and with the decision regions indicated.

Primary decision errors are corrected by the secondary decision when the magnitude of the apparent error (from the incorrect primary decision point to the received signal) is largest in the orthogonal direction of the correct signal. Suppose that the actual error vector, between the transmitted signal and the received signal, is (e_1 , e_2 , e_3 , e_4). When the primary decision is correct, the receiver secondary information designates the largest actual error. When the primary decision is in error, the secondary information is based on the vector between the incorrect primary decision point and the received signal. If the largest error is greater than $2^{1/2}/2$ and less than $2^{1/2}$, and all the other errors are less than $2^{1/2}/2$, the primary decision is in error, but there is a possibility that the secondary decision is correct. The receiver will estimate the error magnitude in the direction of the actual largest error as $\hat{e}_i = 2^{1/2} - |e_i|$, rather than e_i , but the other three error estimates are correct. If the largest error in in the first dimension, the estimated error is $(\hat{e}_1 , e_2 , e_3 , e_4)$, where $|\hat{e}_1| = 2^{1/2} - |e_1|$ is incorrectly estimated. The secondary decision produces the correct signal when the estimated error is largest in the direction of thecorrect signal.

$$|\hat{e}_1| > |e_2|$$

$$|\hat{e}_1| > |e_3|$$

$$|\hat{e}_1| > |e_4|$$

Using $|\hat{e}_1| = 2^{1/2} - |e_1|$, the requirements for a correct decision are

$$|e_1| + |e_2| < 2^{1/2}$$

$$|e_1| + |e_3| < 2^{1/2}$$

$$|e_1| + |e_4| < 2^{1/2}$$

Allowing the largest error to be in the second, third, or fourth dimensions gives three additional equations.

$$e_2 + e_3 < 2^{1/2}$$

$$e_2 + e_4 < 2^{1/2}$$

$$e_3 + e_4 < 2^{1/2}$$

In agreement with the verbal discussion, there is an error if any $e_i > 2^{1/2}$, or if any two $e_i > 2^{1/2}/2$, and there is no error if all $e_i < 2^{1/2}/2$.

If the inequality is replaced by an equality, each equation defines four decision boundaries in two dimensions, one boundary corresponding to each possible selection of signs instead of absolute values. The decision boundaries discriminate between a signal and its four neighbors in some plane, as shown in Figure 7. The six equations correspond to the six different sets of two dimensional planes in four dimensions. Each boundary of the form $e_i + e_j = 2^{1/2}$, for signed e_i and e_j, places one constraint on four dimensional space, and so defines a three dimensional decision region boundary. Taken together, the six equations each with four sets of signs define the 24 three dimensional boundaries of the four dimensional decision region, so that there is one boundary for each neighbor.

The probability of a correct decision can be computed by integration of the four dimensional probability density over the four dimensional decision region. It is assumed that the probability density is independant gaussian with equal variance in each dimension. These computations have not been performed, but related results appear in the literature [14, 15]. The correct decision probability can be underbounded by integrating the probability density over the spherical region of radius 1, the underbound of the decision region. This spherical region is defined by $X^2 = e_1^2 + e_2^2 + e_3^2 + e_4^2 < 1$, so that the variable X^2 has the chi-square distribution [19] and the integral is tabulated [20].

XIX. CONCLUSION

In bandwidth compressive modulation, increased signal energy is used to achieve higher transmission rate. Familiar multilevel amplitude-phase shift keying (APSK) was shown to be an inefficient signal design in higher dimension, compared to the capacity bound. Four dimensional designs were shown to offer some potential improvement, and two new classes of four dimensional designs were introduced. Each class was devised to correct one of the faults of the PASK design. The first class of designs extends the four dimensional cubic lattice throughout the hypersphere bounded by maximum signal energy. The second uses the densest four dimensional lattice, the alternate vertex cubic lattice, to define the signal points. This lattice provides the smallest decision region volume having a fixed minimum signal separation. By analogy with two dimensional results, it is conjectured that alternate vertex lattice signal designs are similar to the optimum four dimensional designs in geometry and performance. At 2 to 3 bits per dimension, the densest lattice designs increase the rate by 0.6 bits per dimension over APSK, with fixed maximum signal energy and signal separation. Alternately, they allow maximum signal energy to by decreased by 3 or 4 dB, with fixed rate and signal separation. The receivers for the two new classes of four dimensional signal designs are very similar to APSK receivers. The receiver for alternate vertex lattice designs requires a small amount of additional information, the identification of the signal dimension and signal direction having the largest estimated error. The new four dimensional signal designs are suitable for the implementation of bandwidth compressive modulation.

REFERENCES

(1) J.J. Spilker, Digital Communications by Satellite, Englewood Cliffs, N.J. Prentice-Hall, 1977, p. 324.

(2) W.J. Weber, III, P.H. Stanton, and J.T. Sumida, "A Bandwidth Compressive Modulation System Using Multi°Amplitude Minimum Shift Keying (MAMSK)," IEEE Trans. Communications, COM-26, pp. 543-551.

(3) G.R. Welti, and R.K. Kwan, "Comparison of Signal Processing Techniques for Satellite Telephony," NTC'77 Conference Record, IEEE 77CH1292-2 CSCB, 1977, pp. 05:1-1 to 05:1-6.

(4) G.R. Welti, "PCM/FDMA Satellite Telephony with 4-Dimensionally-Coded Quadrature Amplitude Modulation, COMSAT Technical Review, vol. 6, No. 2, 1976, pp. 323-338.

(5) _____, "Guide to Modems," Computer Decisions, 1973, pp. 36-40.

(6) C.E. Shannon, "Communication in the Presence of Noise," Proc. IRE, vol. 37, 1949, pp. 10-21,

(7) C.E. Shannon, "Probability of Error for Optimal Codes in a Gaussian Channel," Bell Syst. Tech. J., vol. 38, 1959, pp. 611-656.

(8) J.M. Wozencraft, and I.M. Jacobs, Principles of Communication Engineering, New York, Wiley, 1965, p. 294.

(9) J.M. Wozencraft, and I.M. Jacobs, op. cit., p. 321.

(10) J.R. Davey, "Modems," Proc. IEEE, vol. 60, 1972, pp. 1,284-1,292.

(11) J.M Wozencraft, and I.M. Jacobs, op. cit., pp. 323-332.

(12) J.M. Wozencraft, and I.M. Jacobs, op. cit., pp. 317-320.

(13) G.J. Foschini, R.D. Gitlin, and S.B. Weinstein,"Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise," IEEE Trans. Communications, COM-22, 1974, pp. 28-37.

(14) G.R. Welti, and J.S. Lee, "Digital Transmission with Coherent Four-Dimensional Modulation," IEEE Trans. Information Theory, vol. IT-20, 1974, pp. 497-502.

(15) G.R. Welti, op. cit.

(16) D.M.Y. Sommerville, An Introduction to the Geometry of N Dimensions, New York, Dover, 1958. (Republication of the edition of 1929.) p. 137.

(17) J. Leech, and N.J.A. Sloane, "Sphere Packings and Error Correcting Codes," Canadian J. Math., vol. 4, 1971, pp. 718-745.

(18) D. Slepian, "Permutation Modulation," Proc. IEEE, vol. 53, 1965, pp. 228- 236.

(19) A. Papoulis, Probability, Random Variables, and Stochastic Processes, New York, Mc Graw Hill, 1965, p. 250.

(20) M. Abramowitz, and I.A. Stegun, Handbook of Mathematical Functions, Washington, National Bureau of Standards, 1970, p. 940.

APPENDIX A

REVIEW OF THE TRW

MULTIPLE ACCESS STUDY

This report discusses the "Mobile Multiple Access Study" prepared for NASA-Goddard by TRW, and considers how this effort affects the work planned by NASA-Ames. The TRW study was performed with limited objectives, budget, and schedule, and is generally responsive to Goddard's RFP.

The study objectives were:

1) to consider FDMA, TDMA, and CDMA multiple access techniques, and to select the best based on terminal cost, operating proceedures, system capacity, and satellite complexity and cost,

2) to describe the system using the selected multiple access technique, including hardware parameters and operation proceedures, and

3) to design and estimate the cost of the mobile terminals.

This study is part of the Public Service Communications Satellite development, and several important system constraints are given in the RFP. There is a single channel per terminal, implying single channel per carrier, and continental US coverage is required. The transmitted signals are as follows;

1. Voice: Narrowband FM, BW = 10 kHz, C/N = 47dB-Hz
2. Data: PCM-PSK, 75 and 300 bps, E/N_o = 10.3 dB
3. Fax: 1200 or 2400 bps, E/N_o = 8 dB.

Possible system configurations include;

1. several or multiple fixed beams
2. UHF or L band
3. user to user/ user to central station with connection to an existing network.

The selected multiple access technique is FDMA, because of its simplicity and greater voice channel capacity. FDMA's simplicity is reflected in lower terminal cost and less complex operation. Although the comparison of multiple access techniques is not given in the required parametric form, TRW's reviews of the techniques are of interest. For FDMA, a random access order wire and central station frequency assignments are recommended. As in SPADE, the carriers are voice activated to save transponder power. It seems generally accepted that FDMA is the simplest and most efficient multiple access technique.

TDMA and CDMA require a digital voice modulation, rather than the specified FM.

FDMA is optimum for digital modulation as well as FM, and FDMA/PSK is used in INTELSAT's well known SPADE system. Although SPADE used 7 bit PCM and QPSK, continuously variable slope delta modulation is more nearly competitive with FM, which is considered best. Delta modulation is less sensitive to errors than PCM, and there has been LSI development for both techniques.

The TRW discussion of narrowband FM is somewhat puzzling. TRW uses Carson's rule for the bandwidth of wideband FM, and the signal-to-noise improvement formula for wideband FM, in the discussion of narrowband FM. It is well known that the bandwidth of a narrowband FM signal extends from the carrier to plus-and-minus the highest baseband frequency, exactly as in ordinary AM. Narrowband FM has essentially the same signal-to-noise performance, at high signal-to-noise ratios, as ordinary AM. This is why wideband FM is most frequently used. The apparent reason for using narrowband FM is to reduce voice channel bandwidth in the satellite transponder. The given bandwidth of 10 kHz can be interpreted as twice the highest voice frequency (4kHz), plus a guardband (2kHz). TRW allows 25kHz per voice channel, apparently 10 kHz for the modulated voice and a 15 kHz guardband. It is possible to reduce the transponder guardbands and bandwidth, or to use wideband FM (β =4, BW = 20 kHz), or to use digital modulation.

The study presents a user traffic model, and concludes that 72 multiple access channelscan accomodate 1,200 heavy users or 10,000 light users with one-tenth of the attempted calls receiving a busy signal. 28 channels are permanently assigned to very heavy users. The user model defines a heavy user as one making five calls, of ten minutes duration, in an eight hour day. A light user makes two three minute calls. No actual data are given, and this loading seems optimistic. The call attempt rate parameter, β , is incorrect on page 3-2, and correct on page 3-5.

The FDMA system design has several interesting results. Systems are described using a single beam, four time zone beams, and eight north-south time zone beams. Although the eight beam system accomodates 400 rather than 100 voice channels, the power and weight requirements favor the four beam system. UHF rather than L band is chosen simply for a 6 dB reduction in path loss. There are no linear, space qualified UHF amplifiers available, and TRW has submitted a proposal to develop one, which would use twenty-four transistors in parallel. Using a K_u band downlink to a central

station would save UHF bandwidth.

Block diagrams and cost estimates are developed for FDMA, TDMA, and CDMA mobile ground terminals. The costs for full capability (transmit, receive, multiple access) FDMA, TDMA, and CDMA, terminals are $1,900, $2,500, and $2,500 per unit for 10,000 units. About two-thirds of these costs are for component parts. The quoted costs exclude the R&D and production design cost, which is estimated at 2 to 4 million dollars, or an additional $200 to $400 per unit for 10,000 units.

TRW considers the following topics worthy of further study;

1. steerable mobile antenna
2. spacecraft UHF multibeam antenna
3. 25 kHz transponder filters, surface wave acoustic
4. UHF linear transponder (proposal submitted)
5. automated network control
6. control system design, queing, priorities
7. multipath and RFI
8. passive IM
9. transponder IM
10. multipaction (low pressure electron resonance)
11. K_u , K_a band mobile equipment
12. user traffic model
13. tolerable delay and busy
14. A/J requirement
15. privacy requirement
16. voice quality requirement

In view of the major emphasis on system configuration and terminal hardware in this study, and the fact that FDMA/FM is generally accepted to be superior for large numbers of users, it was useless to wait for these results before issuing the Ames RFP.

The interesting question seems to be, "How close can digital voice approach FM?" While FM is superior in both EIRP and bandwidth requirements, digital voice has useful features that might counterbalance its disadvantages. Assuming that the Public

Service Communications Satellite will have a mobile service transponder designed for FDMA/FM, Ames should investigate the possibility of a compatible digital voice experiment, perhaps using laboratory rather than fully packaged equipment.

There are several reasons that digital voice should be considered;

1. gradual conversion of land line links to digital
2. cost reduction for large quantities by LSI development
3. digital voice compatible with data
4. digital voice easily scrambled for privacy
5. delta modulation more tolerant of marginal signal-to-noise
6. digital can use regenerative satellite repeaters, in later applications
7. digital modulation can be made narrow band by bandwidth compressive modulation

FM is a wideband technique, suitable for the older generation of power limited satellites. Current satellites are bandwidth limited, as indicated by the choice of narrowband FM. If high power is available, bandwidth compressive modulation (eight phase, etc.) can reduce digital bandwidth below that required for narrowband FM.

The first step in planning a compatible digital experiment would be to compare FM, PCM, and adaptive delta modulation parametrically, varying signal-to-noise ratio and bandwidth. FM would be considered at various quality (signal-to-noise) levels and modulation indices, including narrowband. PCM and delta modulation would be considered at various quality levels(bits per sample, error rate) and with different RF modulation methods (QPSK, MSK, eight-phase). The TRW study assumed narrowband FM, and selected FDMA; this study would assume FDMA, and compare voice and RF modulation techniques, for equal subjective voice quality.

The second step would consider the impact of the different modulation methods on system design. Given some fixed path loss and mobile and satellite G/T, link calculations would be performed to determine mobile and satellite power. The effect of transponder intermodulation must be considered for the different modulation techniques. Although the total transponder power and bandwidth would be determined by the FDMA/FM design, it is possible to have a digital experiment at different voice channel power and bandwidth than that of the FDMA/FM design. Several FM channels could be occupied by one digital channel, and digital carrier power could be reduced

by tone loading the transponder. Narrow band, high power digital channels could be used by leaving some of the transponder bandwidth empty.

The last step in designing a digital experiment depends on the final design of the Public Service Satellite transponder. During the earlier steps, consideration should be given to the possibility of designing a competitive digital system. It may be that considerations of privacy, transponder intermodulation, or UHF amplifier design will indicate the need for an approach other than the familiar FDMA/FM.

APPENDIX B

REVIEW OF POWER BUDGETS AND

RECENT PAPERS PERTAINING

TO SATELLITE COMMUNICATIONS

This report will review the power budgets suggested for land mobile communications by several sources, and will briefly describe three recent papers describing voice modulation techniques.

The system configuration of a satellite mobile communication system provides necessary background for the study of modulation and multiple access methods. We should consider if such a system is feasible, if it is EIRP or bandwidth limited, and if modulation and access methods affect feasibility. We will examine the parameters suggested by several sources, including Sam Fordyce of NASA headquarters, the TRW study done for NASA Goddard (and previously reviewed), the STI proposal, and the PSCS brochure published by NASA. The parameters used in a recent demonstration by Dr. James Brown of NASA-Goddard are given for comparison.

Both the up-link and down-link will be in the 800-947 MHz UHF band, and the same antennas will be used for both. Because of the limitation on spacecraft power, the down-link is more critical. Table I gives the down-link budgets from the different sources. Calculated parameters, indicated by parentheses, were found using formulas in the ITT Reference Data for Radio Engineers. The power budgets are similar, due to common assumptions.

The satellite antenna could be 15 ft, providing 30 dB gain and continental US coverage, if 20-30 dBW transponder power is available. The TRW budget shown has an antenna for each time zone, and 2.6 dB more gain. The mobile antenna gain of 3 dB has been a ground rule of these studies, but has been questioned by TRW and STI. High gain mobile antennas would ease down-link power budget problems, but would increase mobile receiver cost. Servo controlled tracking dishes (1.5 ft - 10 dB) or Yagi arrays (0.5 ft - 7 dB) might be workable. A better system might be several directional antennas, with the strongest signal selected electronically, or even an electronically phased array. The antenna cost should be less than a few hundred dollars.

For the modulation used, the bandwidth varies from 10 to 25 kHz. For a 25kHz allocation, including guardband, 4 MHz provides 160 voice channels. The power required for each active voice channel varies from 3.2 to 12 Watts. By removing power from inactive voice channels, as in SPADE, the average power can be reduced 60% or 4 dB, and noise is also reduced. Since 40 to 200 voice channels are required, total transponder power is very large.

Table I: Down-link power budgets from several sources

Budget source	Fordyce, NASA	TRW	STI	Brown experiment	NASA PSCS
Free space loss	183	183.6	181	(189, 1800MHz)[a]	183.6
Frequency	860 MHz[b]	850	1000	L band[c]	866-870
Sat. ant. G	30, 15 ft	32.6	26	(41.7, 30 ft)	29.1
Mobile ant. G	3	3	3	(10, helix; 3, dipole)	3
Mobile rcvr temp	1000 °K	500	700	800	500
Mobile G/T	(-27 dB/°K)	-24	(-25.45)	(-19; -27)	-24
Margin	(3-4 dB)	10.6	6		4.9
Modulation	FM, β =3	NBFM	digital	FM, β =1.5	
Channel bandwidth	25 kHz	(10)	(20) 16 kbps, rate 1/2	15	(100)
FM gain	15 dB	(0)		(5.3)	
C/N_o , E/N_o	(57 dB/Hz)	48	6		66
Channel power	10 W	3.2	12	40	(7.5)
Transponder power	2 kW	32 W	50 W	40 W	300 W

a Parameters which were not in the original budget, and could be computed, are indicated by parentheses.

b Units are the same in each row, unless otherwise indicated.

c For L band, free space loss and satellite antenna gain both increase 6 dB.

The power budgets, and experiments, show that some mobile service can be provided. The question of feasibility reduces to the cost per voice channel. Sam Fordyce estimates the cost of satellite and launch to be $20 million, and operating costs for ten years life at $16 million. The TRW study found that 100 voice channels can accomodate 28 preassigned users with 1,000 heavy demand assigned users or 10,000 light users. Emergency only access or very expensive access would increase the total number of users greatly. There are about 300,000 mobile telephone users and 10 million CB users currently. Commercially available land mobile radio now costs $1,000 per unit, and TRW estimated that satellite mobile units would cost $2,500 if 10,000 units were produced. The cost of the mobile units, for 10,000 total, exceeds the satellite cost. Costs of constructing and operating a central control station have not been estimated.

The large total cost of land mobile stations has apparently dictated the use of a simple land mobile antenna and of the UHF band. It has been noted that a high gain mobile antenna would allow a smaller spacecraft antenna and less transponder power. With two high gain antennas of constant aperture, the net free space loss minus the antenna gains decreases 6 dB as the frequency is doubled. This further eases the spacecraft design, at the cost of higher antenna surface tolerance and more expensive RF equipment. These options appear closed.

Under these constraints, the spacecraft design problem reduces to selecting the proper mix of antenna gain and transponder power to achieve the required EIRP per channel and number of channels. Linear UHF transponders for spacecraft are not currently available, but should be at least as efficient as those in higher bands. A reasonable system should probably use antennas as large have been used, 30 ft, which would gain 6 dB over the system in the first column of table I. Voice activation gains another 4 dB, and the power required for 100 chanels is 100 W. This is similar to the values given in the second column, by TRW. An adequate system is feasible.

The limited availability of UHF bandwidth and the engineering limits on EIRP provide two limits on the system. The power budgets indicate that both FM and digital techniques provide the required voice channel power and bandwidth.

We next consider three papers that consider modulation techniques for land mobile voice communication.

Bruce Lusignan of Stanford has just completed a study of modulation and voice processing for land mobile radio, funded by an FCC UHF task force. FM and SSB are compared. The commercial land mobile system is FM, with β =1.7. The voice bandwidth is 16 kHz, and channel spacing is 20 to 25 kHz. Acceptable voice quality is achieved at 15 to 30 dB output SNR, with 30 dB the target. Using the current system as a starting point, it was found that sylabic amplitude companded and frequency compressed SSB can reduce voice bandwidth to 1.7 kHz, and channel separation to 2 - 2.5 kHz, a 10:1 reduction. In the 15-30 dB SNR range, the peak power of SSB is about equal to FM, and the average power is about 6 dB less.

Sylabic companding uses a nonlinear amplitude transfer function to reduce the natural amplitude variations of speech by 2 to 1 on a log scale. Companding reduces the variability of the speech SNR, and gives an apparent 15 dB improvement in the system tested.

The speech compression system folds the higher frequencies over the lower frequencies, reducing power and bandwidth to 60% of that normally used. Because the human voice usually produces either low frequency vowels or high frequency consonants, high quality voice results. This is a new invention of Harris, Cleveland, and Lott, and uses straightforward analog circuits. There is no description of this system, and the method of distinguishing the higher and lower frequencies is not obvious. Possibly two quadrature modulated SSB signals are used.

Both amplitude companding and frequency compression are done at baseband, and can be used with any modulation. The gains of amplitude companding are larger for SSB than FM at the low SNR's used in land mobile radio. The study concludes that SSB should be used because of its lower bandwidth requirement. The special SSB equipment would add $100 to $300 to the current installed land mobile unit cost of $1,000. The FCC introduction to this study mentions that there are requests for four or five times the available spectrum in the 860 to 947 MHz band.

At NTC 77, Welti and Kwan of COMSAT compared voice signal processing, modulation, and multiplexing methods for satellite telephony. Although SCPC was excluded, the study is useful. This paper includes the familiar FM and digital methods - FM, companded FM (CFM), 4 and 8 phase PSK, PCM, adaptive PCM (ADPCM), delta modulation (DM) - along with some less familiar methods. Nearly-instantaneous-companding (NIC) is a

is a method of adaptively scaling PCM samples. Variable slope delta modulation (VSDM) is reported to give good speech quality at 24 to 32 kbps and 10^{-3} error probability. In two-pulse amplitude and phase modulation (2P-APM), two separate pulses are transmitted for each sample. Four dimensional quadrature amplitude modulation (4D-QAM) also uses two pulses, but has discrete amplitudes and phases and is designed in four dimensions.

FDM and TDM multiplexing methods were considered, both with and without time assigned speech interpolation (TASI) for FDM and digital speech interpolation (DSI) for TDM. For FDM, the required C/N_o for CFM, 2P-APM, PCM/PSK, or PCM/4D-QAM is about 60 dB/Hz. 2P-APM has the narrowest bandwidth, with CFM and PCM/4D-QAM requiring 50% more bandwidth and FM and PCM/PSK requiring up to 100% more. Some interesting techniques were considered only for TDM, and not FDM. PCM/NIC, DM, and ADPCM require 4 dB less C/N_o than CFM or PCM. DM and ADPCM could operate satisfactorily with a further 2 dB power reduction. For these types of modulation with PSK, the bandwidth is about the same as 2P-APM, the narrowest technique studied for FDM. 4D-QAM reduces the bandwidth another 50%.

Although the results of Welti and Kwan are not for SCPC, they seem to indicate that digital methods are quite competitive. The use of four dimensional, two pulse, signal design for bandwidth compression is interesting, and is similar to a previous suggestion for research at NASA-ARC.

Campanella, Snyderhoud, and Wachs compared FM, CFM, and DM for SCPC in the March 1977 special issue of the IEEE Proceedings on satellite communications. This article is important to the current study to be awarded by NASA-ARC, and GE and Ford reference it in their proposals.

The paper discusses SCPC/FM performance in detail. A PLL receiver extends the FM threshold about 2.5 dB at β 3. Pre-emphasis/de-emphasis gives a net improvement of about 5 dB. Sylabic companding provides gains from a few dB for wideband FM (β =9.3) up to 20 dB for narrowband FM (β =2.7). Because the compandor is not instantaneous, there is a noise burst after each speech burst. This "hush-hush" noise is estimated to cause a 6 dB degradation. These values were used to compute the theoretical FM performance.

SCPC/DM performance was also calculated, with BPSK and QPSK modulation. Digitally controlled-slope delta modulation (DCDM) is used, rather than continuously variable

slope delta modulation (VSDM). Both methods adaptively control the delta modulation step size.

Subjective tests were made by comparing degradations to those produced by Gaussian noise. The experimental results had reasonable agreement with theoretical results. The experimental curves of subjective noise vs C/N_o for CFM and DCDM are very close. The theoretical performance of CFM and DCDM is also quite close, as shown by a table of the theoretical C/N_o at nearly equal bandwidths and a fixed noise level.

Campanella et. al. conclude that, because performance is similar, the choice between CFM and DCDM can be made based on other factors. DCDM is somewhat better than CFM in intermodulation suceptability. The DCDM voice digitizer is not suitable for data transmission, but the BPSK or QPSK modulator can be used directly. DCDM is better than CFM in speech burst detection for speech interpolation, and has greater immunity to carrier frequancy jitter.

These results confirm the conclusion of the review of power budgets, that FM and digital modulation could both provide the required power and bandwidth for a satellite land mobile system. However, both power and bandwidth constraints are tight enough to make the system feasibility marginal. The Lusignan study raises the question of SSB modulation, which has both reduced average power and the minimum possible bandwidth. As SCPC requires a linear transponder, SSB might be useable.

STUDY OF COMMUNICATIONS

DATA COMPRESSION METHODS

FINAL REPORT

Contract NAS2-9703

HARRY W. JONES

Februrary, 1979

Prepared for

NASA AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA, 94035

NASA TECHNICAL MONITOR

LARRY B. HOFMAN

COM-CODE, INC.
305 EASY ST. NO.9, MTN VIEW
CALIFORNIA, 94043

CONTENTS

SECTION A

SUMMARY REPORT

This section, A, briefly describes the contents of this final report for contract NAS2-9703. A previous final report, now designated the interim final report, was delivered in July, 1978. This earlier report described research in transform conditional replenishment, Landsat image compression, and satellite communications. The contract was first modified to include additional research in these three areas and in the new area of simple systems. The contract was later modified again to delete the additional research in Landsat image compression and in satellite communications, and to have reduced additional effort in conditional replenishment.

This report contains four further sections. Section B contains the study plan, systems survey, and literature search results for the simple systems study. Section C describes an investigation into the performance gain for nonstationary image data. The results bound the performance of simple systems. Contrary to the usual assumption, the gain due to nonstationarity is small. Most of the performance gains of adaptive, variable rate compression systems can also be obtained for stationary data. Section D describes the computer programs and the compressed video data available on the SEL 32. The conditional replenishment system that was simulated under this contract was described in the interim final report, and the simulation program used is a SEL 32 version of the previous program. Section E consists of the published versions of two papers that were included as typed versions in the interim final report.

Additional work in video compression should be done at NASA-ARC. Three specific areas are simple systems, conditional replenishment hardware, and publications.

The question of the cost-effectiveness of video compression was considered in the proposal for the additional statement of work, July 28, 1978. Only a few points will be repeated here. Theoretical knowledge and cost of hardware for video compression define several distinct systems, with different cost, transmission rate, and quality. Conditional replenishment has the highest cost, and the lowest transmission rate (1/4 to 1/2 bits per pel) for acceptable quality. Other systems may be more cost effective. An intraframe compressor with variable and limited buffering has lower hardware cost , and medium rate (1 to 2 bpp)

for acceptable quality. The least expensive fixed-rate intraframe compressors, such as adaptive delta modulation, provide acceptable quality at higher rates (2 to 3 bpp). These systems are all compatible with digital links, error correcting coding, and encryption, but the simpler systems would not be major cost components. Simulation and development of simple systems is desireable.

The simulated conditional replenishment system is buildable and effective, but modifications can be made. The I and Q color signals are not tested for mode determination or for changes in time, but simulation artifacts make it obvious that the hardware should make these tests. An extensive range of modes, quantizations, and change thresholds has not been simulated, since they will be optimized in hardware. Some of the features of the simulation are arbitrary, and can be changed. Examples include the refresh by lists, and the absence of directed refresh, for recent changes with lower quality. Fundamental changes, which may introduce unanticipated artifacts, should be simulated.

The NASA-ARC video compression project has made many deviations from the direct approach of simulating, designing, fabricating, and testing the conditional replenishment hardware. The only justification for these diversions is that they improve the final product. Conditional replenishment hardware must be built and demonstrated.

The current video simulation has not been presented in a paper, although an earlier system was described and video tapes presented at two conferences. If new video material of suitable length and quality can be obtained, a paper should be given. Other potential subjects for papers include the quasi-cosine transforms, Landsat table look-up compression, nonstationarity of video data, and bandwidth compressive modulation.

SECTION B

SIMPLE SYSTEMS STUDY PLAN

AND LITERATURE REVIEW

CONTENTS

I. Simple Systems Study Plan

Task II of the additional statement of work is to review the simple systems described in the literature, and to simulate some of the most promissing. The responsive proposal indicated that an inexpensive compressor would use only intraframe compression, and stated that it was unlikely that any inovation could be found in the well studied area of predictive or DPCM compression. The proposal also noted that combinations of two different methods, such as hybrid or dual mode, were often superior to single methods.

The literature review for simple systems has indicated an interesting approach to the design of compression systems. The non-stationarity of video data is a well known design problem. Various ad-hoc design methods have been developed to deal with non-stationarity, such as adaptive predictors or selected quantizers, but the basic design is usually made for an assumed average stationary source. Berger's work on composite sources (1), and the developement of universal coding by Ziv (2), Davisson (3), and others suggest that compression systems be designed specifically for non-stationary data. Since the source statistics vary widely, the compressor should include widely different techniques. This theoretical work explains the effectiveness of earlier systems combining different methods, and suggests further developement in this direction.

The objective of the simple systems study is to devise a highly effective intraframe compression system, using little memory. The approach is to use the non-stationarity, by combining several different basic techniques. The use, and refinement, of Berger's model should make this work new and interesting.

The simple systems study consists of three phases. The first is a general literature review, the second is theoretical and experimental work leading to a video source model, and the third is the selection and simulation of various candidate systems.

The literature review is divided into three topics, systems, universal coding, and non-stationary sources.

II. Basic Compression Systems

In this section, basic compression methods are reviewed from the point of view of complexity and potential performance. The purpose of a teleconference video compressor is to reduce transmission rate as much as possible, while maintaining the desired quality. This is accomplished by removing redundancy from the data, and by introducing unobjectionable distortion. Greater compression requires greater complexity, measured in memory size and number of computations. Since the video image samples have correlation in the scan line, between lines in a frame, and between frames in time, the potential compression can be increased by increasing the memory span to include previous samples, lines and frames. Since the video source is nonstationary, requiring varying rate for constant distortion, a long memory span is also useful for trading bits within a frame to maximize overall quality.

The amount of data required for the compression computation is an approximate indication of system complexity. Various general compression systems are listed in Table 1, according to the compression data span. This table includes most of the different basic techniques in the literature. The major headings 1, 2, and 3 indicate wether the data span is samples, lines, or frames. The subheadings, A,B,C,D,E, indicate the different constraints placed on the data span, single sample, past samples, variable sample span, fixed block sample span, or moving fixed length sample span. Single sample systems can not remove any redundancy, and are used after other compression techniques. For example, transform coefficients are usually quantized with reduced range, and DPCM values are quantized or Huffman coded. Past sample in-line systems, such as DPCM and delta modulation, are the simplest systems, and have been studied since the 1950's. Variable sample span systems are also conceptually simple, but produce variable output rate per sample. The fixed block techniques are more powerful than past sample techniques, but often introduce block edge artifacts. Like fixed block techniques, moving window techniques introduce delay so that future samples are used, but edges are avoided.

Table I. Compression methods classified by data span.

	1. In-line compression	2. Intraframe compression	3. Interframe compression
A. current sample only	quantizers Huffman coding universal coding classification		
B. past samples	DPCM delta modulation dual mode predictors	DPCM delta modulation dual mode predictors line repeat	DPCM delta modulation dual mode predictors field/frame repeats
C. variable sample span	run length coding interpolation adaptive sampling constant area	contour blob	conditional replenishment motion prediction
D. fixed block length	transform polynomial transform highs/lows	transform highs/lows bit plane	transform
E. moving fixed length	digital filters convolutional codes analog filters	digital filters line averaging line interpolation	temporal filter frame averaging frame interpolation

These techniques can be extended to several lines or several frames. Some extensions are familiar, such as two and three dimensional DPCM and transforms. Line and frame repeat are degenerate cases of differencing or transforming, in which the difference of needed coefficient data are not transmitted. Contouring is the two dimensional analog of identifying line sample groups with similar values, as in run length coding. The three dimensional or time analogs are conditional replenishment or motion prediction, which identify two dimensional regions that persist for some variable time span.

Table 1 emphasizes distinctions made using the data span required. The three vertical divisions indicate the amount of data in memory, while the horizontal divisions indicate the data span used in the compression calculation. These are two basic indications of system complexity. The memory requirement is equal to the data span required in future computations, while the computation complexity is proportional to the data span used in each calculation, for example the transform block size. Similarly, the data span is an indication of potential performance. The use of many previous in-line samples gives little gain over the use of one previous sample, but the use of samples in adjacent lines and frames gives significant additional compression. The brute force approach to increasing compression is to include more redundancy in the computational data span. Even when additional lines and frames are included in the memory span, the useful computational span remains local.

There are several other important bases for classification of compression systems, including fixed rate/variable rate, and fixed method/adaptive method. It is realized that the above discussion of different systems is very brief. Further discription of basic compression systems will be given as part of the selection of the systems to be simulated.

III. Universal Coding

The theory of universal coding is useful in the compression of video data, even though video data is nonstationary, and universal coding usually assumes a stationary source. Davisson and Grey (4) have written an excellent review article. The classical Shannon theory defines the rate distortion bound for source compression, when the source statistics are known. Universal coding considers the problem of encoding any one of a class of sources, each with different statistics. Many of the proofs of universal coding theorems are constructive rather than existance proofs, and give insight into the design of practical systems.

Universal coding techniques are usually classified as fixed rate or variable rate, and as noiseless or having a fidelity criterion. We first consider fixed rate universal coding. Sakrison (5) found that, if the average distortion is constrained for all possible sources, the fixed rate required for encoding an unknown source is the largest rate that would be required for any possible source. For fixed rate source coding and bounded distortion, the required rate is that of the worst case source. Ziv (2) examined the distortion of fixed rate systems, and found that universal codes can be constructed to achieve the mimimum attainable distortion for the true but unknown source. For fixed rate source encoding, Sakrison showed that the rate must be sufficient to encode the worst source with the allowed distortion, but Ziv showed that the distortion for any source need be no worse than if the exact source statistics were known. Universal codes are those that achieve a bound for known statistics, even when the exact statistics are not known in advance. Neuhoff,Grey, and Davisson (6) present a unified theory of fixed rate universal coding. Different definitions of universality, and different conditions on the class of sources are considered.

Davisson (3) gave a full treatment of universal noiseless coding, and first defined the different types of universality. Ziv's universal codes can result in a chosen fixed rate that is greater than the message entropy (the rate required for noiseless coding) for the actual source. Using the difference in noiseless transmission rate, Davisson showed the existance of variable rate universal codes that achieve the minimum rate for noiseless coding for any unknown source in a class of sources. For fixed rate noiseless coding, the transmission rate must

be greater than the entropy of the worst source, as in the fixed rate, bounded distortion result of Gakrison.

Variable rate universal coding with a distortion criterion has been examined in more recent papers by Davisson, Pursley, Makenthun, and Keiffer (7,8,9,10). The different types of universality first introduced by Davisson are considered. Universal codes exist which approach the minimum rate for the actual source selected, while not exceeding a fixed distortion.

Grey and Davisson (11) give a simple universal coding theorem. Suppose there is a class of sources producing symbols in some alphabet, so that each source has different symbol statistics. The number of possible sources in the class, m, is finite, and one of the sources is selected initially. The source coder reproduces the source alphabet symbols using some reproduction alphabet. A distortion measure defines the average distortion between the source symbols and the reproduction symbols, and each possible source can be coded according to its own rate distortion bound. That is, for any distortion, D, there exists a code book with rate less than or equal to the rate distortion bound R(D), if the number of symbols, n, in a coded block is sufficiently large. Since the number of possible sources is m, the codewords in each source codebook can be augmented using $\log_2 m$ bits to indicate which source codebook they belong to. The source coder encodes a block of n symbols using the shortest augmented codeword in any codebook. The rate is increased by $n^{-1} \log_2 m$ bits per symbol, and this increase approaches zero for large n. Thus the rate distortion bound, which is defined for large n, is achieved for any possible source.

The uncertainty about the actual source requires added information transmission, which causes the rate to approach the rate distortion bound more slowly as n increases. If the class of sources is infinite, as it would be if indexed on a continuous parameter, the distortion measure can be used to divide the class of sources into subclasses, such that each can be coded using a single code that nearly achieves the rate distortion bound for all members of that subclass. If the number of subclasses is finite, the above proof holds.

The proofs of universal coding theorems all use a similar fundamental approach. The message describing a source symbol block is divided into two parts. The first part indicates the method of encoding, and the second part contains the

encoded data. The method of encoding is indicated in two different ways. In constructive proofs, the source statistics needed to define the coding method are estimated and transmitted. For example, the probabilities of the codewords define the Huffman code. Existance proofs are similar to Shannon's original proofs of the channel-coding/capacity theorem and of the source-coding/ rate-distortion theorem, in that average or typical codes are shown to achieve the bounds, but no construction method is given. The theorem proof outline given above is an existance proof, because good codes for each possible source were assumed to exist, and used in the universal code. In both constructive and existance proofs, the overhead of the first part of the message, used to indicate the coding method, is negligible for large source symbol blocks.

The universal coding constructive proofs use some interesting ideas. An early method was suggested for run length compression by Lynch (12) and Davisson (13), and was later considered in more depth by Cover (14). Suppose that only a small number of the original data samples in a fixed block are to be transmitted. This information can be transmitted as the number of samples used, the location of the samples, and the value of the samples. The number and values are directly encoded, while the location is encoded as the list ordering of the correct permutation of q samples in n possible positions. If the alphabet size is p, the transmission rate for q symbols transmitted out of a block of n is

$$R = n^{-1} \left(\log_2 q + \log_2 \binom{n}{q} + q \log_2 p \right)$$

The first term is negligible for large n, the second term describes equally likely messages which have no potential coding gain, and the third term indicates a sample reduction compression gain of q/n.

The histogram method of Fitinghoff was described by Davisson (3), and was developed before universal coding. The conditional histogram of data subblocks is measured, encoded, and transmitted, along with the codeword for the full block constructed according to the histogram. If the source symbols are independant, the histogram does not have to be conditional. For example, the probability of each symbol can be transmitted and used to define a Huffman code for blocks of symbols.

Ziv (2) used a different method to obtain the first universal coding results. A block of n symbols is broken up into n/k subblocks of k symbols. The first I different subblocks are directly encoded and transmitted, using $k(\log_2 p)I$ bits, where p is the alphabet size. These I subblocks are used to construct a table, and subsequent transmissions indicate the subblocks by their position in the list, using $\log_2 I$ bits. The total rate is

$$R = k (\log_2 p)\ I + (n/k) \log_2 I$$

This method causes errors if the list does not contain all the required subblocks, but this is improbable for large I and n. The effect of errors is reduced if the closest list subblock is used, or if the table is continually revised.

IV. Nonstationary Sources

The coding of nonstationary sources is a practical problem that has received little theoretical attention. Berger developed a fundamental model for nonstationary sources (1). A composite source is a finite class of individual sources, one of which is selected by a statistical switch. The switching probabilities determine the relative amount of time each source is used. If neither the encoder or the decoder can be informed of the exact sequence of switch positions, the composite source is an unseparable mixture of the individual sources. The average fraction of each source used, and the symbol statistics of each source, define the statistics of the mixture distribution. The only thing that can be done is to encode the composite source as one stationary source, having the mixture statistics. However, if both the encoder and the decoder can be fully informed of the sequence of switch positions, the rate distortion bound of each individual source can be achieved. If the sources have very different statistics, the average of the rate distortion bounds is more favorable than the rate distortion bound of the mixture source. This model will be further investigated and applied in later sections.

Ziv (2) extended universal coding to the problem of encoding any one of a class of nonstationary sources. This is not the the problem of encoding one nonstationary source using Berger's model. A fixed rate, unchanging code is assumed. It is assumed that each nonstationary source is an unseparable mixture composite source. Ziv finds fixed rate universal codes which achieve, for any selected source, the minimum distortion obtainable by any fixed rate code for the exact nonstationary source statistics. There are universal codes that achieve the mixture statistics rate distortion bound, for any one of a class of nonstationary sources.

Grey and Davisson (11) consider a model similar to that of Berger. The composite source can be considered to be locally stationary, while the switch is in one position. If the switch position varies very slowly, then the universal coding results for a permanently fixed switch can be applied to the nonstationary source. The code block length should be short in comparison to the average switching time. Grey and Davisson later (4) reemphasize that the correct choice of block length allows universal coding for nonstationary sources that are locally stationary.

Grey and Davisson (4) coded video data (apparently Landsat) using DPCM with a fixed length code, three different Huffman codes, and a run length coder. Although the average sample entropy of the differences was 3.3 bits, selecting the best coding method for each block of 64 samples gave a rate of 3 bits per sample for noiseless coding. A rate distortion bound less than the mixture rate distortion bound was achieved, because of the effective separation of the subsources of a nonstationary source. Rice and Plaunt (13) earlier used a system of three different Huffman codes on similar data, and considered their results to be near optimum because they were within 0.25 bits per sample of the mixture entropy. More recent authors still appear to consider the mixture rate distortion function to be the appropriate bound for nonstationary sources.

The above work of Berger and of Grey and Davisson provides a new basis for the compression of nonstationary sources. Nonstationary sources can be modeled as switched subsources, and the rate distortion of any subsource can be approached, if the overhead to indicate subsource switching is negligible. Since the rate distortion bound of a separable composite source can be less than the rate distortion bound of the corresponding mixture source, the consideration of nonstationarity provides an opportunity to improve performance.

REFERENCES

1. T. Berger, Rate Distortion Theory,Prentice-Hall, Englewood Cliffs, New Jersey, 1971, ch. 6.

2. J. Ziv, "Coding of Sources with Unknown Statistics-Part I: Probability of encoding Error," and "...-Part II: Distortion Relative to a Fidelity Criterion," IEEE Trans. Inform.Theory, vol. It-18, pp.384-394, May 1972.

3. L. Davisson, "Universal Noiseless Coding," IEEE Trans. Inform. Theory, vol IT-19, pp. 783-795, 1973.

4. L. Davisson and R. Gray, "Advances in Data Compression," in Advances in Communications Systems, vol. 4, A.V. Balakrishnan and A.J. Viterbi, editors, Academic Press, New York, 1975, pp. 199-228.

5. D.J. Sakrison, "The Rate Distortion Function for a Class of Sources," Information and Control, vol. 15, pp. 165-195, 1969.

6. D.L. Neuhoff, R.M. Gray. and L.D. Davisson, "Fixed Rate Universal Block Source Coding with a Fidelity Criterion," IEEE Trans. Inform. Theory, vol IT-21, pp. 511-523, Sept. 1975.

7. M.B. Pursley and L.D. Davisson,"Variable Rate Coding for Nonergodic Sources and Classes of Ergodic Sources Subject to a Fidelity Constraint," IEEE Trans. Inform.Theory, vol. IT-22, pp. 324-337, May 1976.

8. M.B. Pursley and K.M. Makenthun, "Variable Rate Coding for Classes of Sources with Generalized Alphabets," IEEE Trans. Inform. Theory, vol. IT-23, pp. 592-597, Sept. 1977.

9. J.C. Keiffer, "A Generalization of the Pursley-Davisson-Mackenthun Universal Variable-Rate Coding Theorem,"IEEE Trans. Inform. Theory, vol. IT-23, pp. 694-697, Nov. 1977.

10. K.M. Mackenthun and M.B. Pursley, "Variable-Rate Universal Block Source Coding Subject to a Fidelity Constraint," IEEE Trans. Inform. Theory, vol. IT-24, pp. 349-360, May 1978.

11. R.M. Gray and L.D. Davisson,"A Mathematical Theory of Data Compression?", Proc. 1974 Intern. Conf. Commun, pp. 40A-1-40A-5.

12. T.J. Lynch,"Sequence Time Coding for Data Compression," Proc. IEEE, vol. 54, pp. 1490-1491, 1966.

13. L.D. Davisson,"Comments on 'Sequence Time Coding for Data Compression' ", Proc. IEEE, vol. 54, p. 2010, 1966.

14. T.M. Cover,"Enumerative Source Encoding," IEEE Trans. Inform. Theory, vol. IT-19, pp. 73-77, 1973.

15. R.F. Rice and J.R. Plaunt, "Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data," IEEE Trans. Commun. Tech., vol. COM-19, pp. 889-897, Dec. 1971.

SECTION C

COMPRESSION OF

NONSTATIONARY IMAGE DATA

CONTENTS

ABSTRACT

It is well known that the video image source is nonstationary, and that adaptive compression can obtain improved performance. In this paper, a composite source model for nonstationary sources is developed. This model illustrates how the improved rate distortion bound for nonstationary sources is used in practical design. All the source models assume that the intersample dependancy of the video samples is removed by a first order one dimensional predictor. The experimental statistics of a test set of video images are examined to define the parameters of different nonstationary source models. The performance of adaptive prediction, adaptive entropy encoding, and adaptive quantization are examined and compared to results reported in the literature. It is found that the improvement in rate distortion bound due to nonstationarity is relatively small, and much of the gain of fixed rate adaptive systems is achieved by more closely approaching the rate distortion bound for stationary data. In agreement with quantization theory, variable rate entropy encoding of quantizer output values closely approaches the rate distortion bound.

INTRODUCTION

It is shown in information theory that memory reduces the information rate of a source. A source with memory can be modeled as a composite source, having different subsources incorporating the effect of different past symbol sequences on the next symbol. At any given time, one of the subsources is selected by a switch. When the rate required to transmit the switch position is included, the transmission rate of the composite source is equal to the average rate of the subsources, plus the additional rate to exactly indicate the switch position. The rate distortion bound is defined by the optimum selection of the subsource definition, and of the optimum source coding for each subsource. The composite source model demonstrates the relationship between information theory and the common adaptive source coding method of first selecting the best of several alternate encoders and then identifying the particular encoder to the decoder using overhead transmission rate.

The composite source model is used in the analysis of adaptive source coding for the video image source. The memory due to the nonstationarity of the Markov model of sample dependance is investigated, rather than the memory due directly to the sample dependance. If the general form of the data distribution is known, the first order Markov model for the dependant video samples is defined by the mean, variance, and correlation of the samples. These determine the design of the predictor, quantizer, and quantizer output entropy encoder. The experimental mean, variance, and correlation obtained using wide area averages define the image mixture source, which is the stationary source having the same subsources and subsource probabilities as the actual nonstationary source. The rate distortion function of the mixture source is an upper bound on the actual rate distortion function, and the mixture source statistics restrict the composite source model.

Analysis of the mixture source statistics indicates that adaptive predictors can not obtain any significant improvement in rate distortion performance. This is confirmed by several experiments. The adaptive image encoders described in the literature all use adaptive entropy encoders or adaptive quantizers.

Experimental results on adaptive entropy encoding indicate that the potential gain is less than ten percent, for one test set of video images. Such gains have been reported previously for Landsat images. Experiments with adaptive quantizers are also in agreement with reported results that rate improvements of about twenty-five percent can be obtained. Theory indicates that much of the gain of adaptive quantizers is due to more closely approaching the rate distortion bound for the equivalent stationary source, with ten percent due to the nonstationary memory.

THE SWITCHED SUBSOURCE MODEL FOR SOURCES WITH MEMORY

Statistical data dependancy, due either to correlated data or to nonstationary data statistics, reduces the information transmission needed to achieve a specified fidelity. In this section, the switched subsource model for sources with memory is described. The rate distortion function (rdf) of a source is the minimum transmission rate required to achieve a given fidelity. The switched subsource model shows how an improved rdf can be achieved by the identification of subsources, which incorporate memory of the previous data. The source model is based on Berger's composite source with side information [1], and is fundamental in adaptive source compression and universal coding. The switched subsource model will later be developed using experimental image data, and used to design adaptive image compression systems.

A composite source, as defined by Berger, is shown in the left part of figure 1. The boxes contained in the composite source, labeled P(x/s), s = 1, 2, ... M, are independant subsources which produce discrete symbols in the common alphabet x_i, i = 1, 2, ... A. At any given time, one subsource is selected by a switch with known statistics. As in Berger's work, the rate distortion bound of the composite source is determined by examining the performance of a source encoder/decoder, which is also shown in figure 1. The switch position is determined by direct observation, or by examination of data which is delayed and encoded after the switch position is estimated. The switch position is described to the source symbol encoder, which uses the optimum encoder for each subsource. The switch position information is then combined with the encoded symbols, and transmitted to the decoder. The decoder uses the switch position information to select the correct source symbol decoder.

Berger considered cases where the switch positions are independant random variables, or are arbitrarily controlled. Here, the case of a switch with memory is also considered. Berger considered cases where the side information describing the switch position was provided directly to the symbol encoder and decoder. Here, the overhead rate required to transmit the switch position information is included. Berger's results include limiting cases of the model of figure 1.

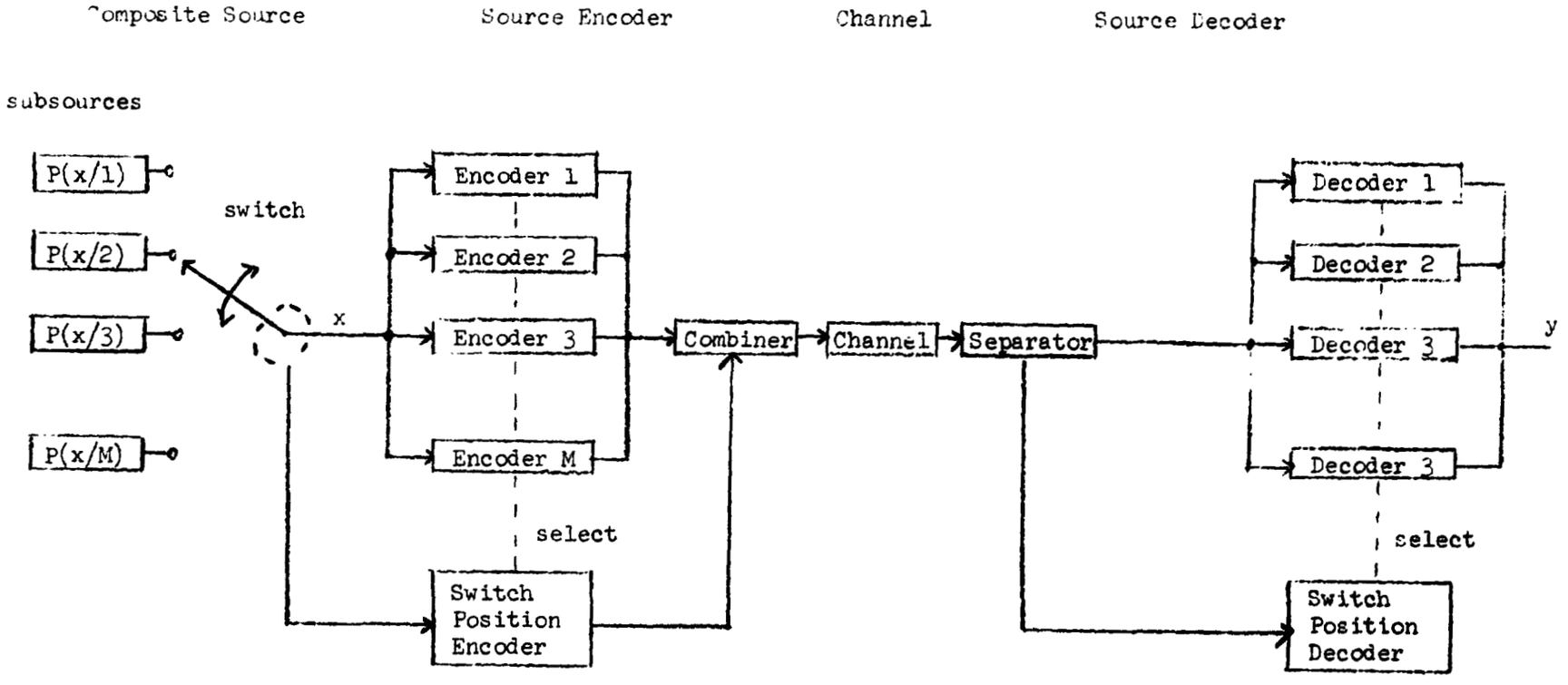


Figure 1. The composite source model and source encoder and decoder.

The rate distortion function of the composite source is the minimum of the mutual information of the source encoder and decoder of figure 1. x is the source symbol, y is the corresponding decoded symbol, and their joint probability distribution, P(x,y), is determined by the encoder/decoder. The mutual information between x and y is

$$I(x;y) = \sum_{i,j} P(i,j) \log_2 \frac{P(i,j)}{P(i)P(j)} \qquad (1)$$

The indices i,j vary over all combinations of source symbol x and receiver symbol y.

The subsource identification information is used by the encoder and decoder to select a particular encoder and decoder. The joint distribution of x and y for the subsource k is P(x,y/k), and the mutual information is

$$I(x;y) = \sum_{i,j} \sum_{k} P(k)\, P(i,j/k) \log_2 \frac{P(k)\, P(i,j/k)}{P(k)P(i/k)P(k)P(j/k)}$$

$$= \sum_{i,j} \sum_{k} P(k)\, P(i,j/k) \left(\log_2 \frac{P(i,j/k)}{P(i/k)P(j/k)} - \log_2 P(k) \right)$$

$$= \sum_{k} P(k) \sum_{i,j} P(i,j/k) \log_2 \frac{P(i,j/k)}{P(i/k)P(j/k)}$$

$$- \sum_{k} P(k) \log_2 P(k) \qquad (2)$$

If d_k is the distortion of the k th subsource, the average distortion for the composite subsource is

$$D = \sum_k \cdot(k)\, d_k$$

The rate distortion function is the minimum of the mutual information for all subsource encoder/decoders, or all P(x,y/s), such that d_k is achieved for each subsource.

$$R(D) = \min_{\substack{P(x,y/s) \\ d_k}} I(x;y)$$

$$= \sum_k P(k)\ R_k(d_k) \quad + \quad H_{sw}[P(k)] \qquad (3)$$

where the d_k are such that the equation for D is satisfied, and the slopes of the $R_k(d_k)$ are equal (see Berger [1] pp. 184.55). $R_k(d_k)$ is the rate distortion function of subsource k, and $H_{sw}[P(k)]$ is the entropy of the switch, for independantly chosen switch positions. The first term of equation 3 corresponds to a result of Berger. The equation indicates that the source encoder/decoder of figure 1 achieves the rate distortion function of any selected subsource, at the cost of exactly transmitting the subsource identification.

Suppose that, instead of being independant, the switch position during each symbol depended on the previous switch positions. Then P(i,j/k) can be replaced by P(i,j/k,l) in the above derivation, where l describes the effect of previous switch positions. Since the current encoder/decoder depends only on the current switch position, the above result is obtained with $H_{sw}[P(k)]$ replaced by $H_{sw}[P(k,l)]$. $H_{sw}[P(k,l)]$ is the entropy of a switch with memory. A switch with memory models sources where the switch position changes slowly compared to the symbol rate.

The rdf of a composite source is the minimum mutual information, under the constraint on the distortions d_k, for all subsource identification methods and for all encoder/decoders. To determine the rdf, the switch position identification method must be variable, although the the composite source definition is unchanged.

$$R(D) = \min_{\substack{P(x,y/s) \\ d_k \\ sw}} I(x;y) \qquad (4)$$

The only change from equation 3 is the indication, "sw", that the minimization is made over all methods of switch position indication, as well as all encoder/decoder sets, and the constrained distortion. The switch position indication method may vary in different regions of the rdf. If the switch position indication can be transmitted at no cost, the fullest possible subsource identification information should be transmitted. The subsource definitions are often arbitrary to some extent, and the composite source can be modeled as a unifilar source, which has each source symbol uniquely associated with a subsource [2]. For a unifilar source, all the source symbol information is transmitted as subsource identification information, so that some constraint on this information is necessary.

When no subsource identification is transmitted, a single encoder/decoder must be used for all the subsources. As indicated by Berger, the composite source is treated as single source, having the mixture source probability distribution.

$$P_{mix}(x) = \sum_k P(k)\, P(x/k)$$

The mixture statistics determine the mutual information and the rdf.

$$R_{mix}(D) = \min_{\substack{P(x,y) \\ D}} I(x;y) \qquad (5)$$

where $I(x,y)$ is given by equation 1. Obviously,

$$R(D) \leq R_{mix}(D)$$

If $R(D) = R_{mix}(D)$, the rdf is not improved by using the subsource identification information, and the source is effectively stationary. When $R(D) < R_{mix}(D)$, the use of separate encoder/decoders for each subsource results in a improved rdf, even when the required subsource identification overhead transmission rate is considered.

DISCUSSION OF THE SWITCHED SUBSOURCE MODEL

The composite source model of figure 1 can be used to represent a source with dependant samples, by including one subsource for each possible current state, as defined by the relevant past source symbols. The model can also be used to represent a source with a variable symbol distribution, by including one subsource for each distribution. The different subsources represent different past source behavior, which affects the current symbol output. Removing the sample dependancy is the basic method of data compression. Adaptive source coding techniques, based on a nonstationary source model, usually achieve superior compression performance. In this paper, the composite source model is used to investigate the nonstationary behavior of correlated image data.

In the image coding literature, many adaptive systems have been developed which change the encoder/decoder to optimize performance for nonstationary signals. Rate reductions of one-third to one-half have been achieved, compared to the best single encoder/decoder [3]. Usually, the method of figure 1 is used. Experimental insight is used to partition the nonstationary source into subsources, an encoder/decoder is designed for each subsource, and the subsource identification and the encoded symbols are transmitted. In an early paper using this approach, Tasto and Wintz [4] derived the rate distortion bound of equation 3, which they presented as an experimental upper bound on the rdf. For their image data and particular subsource definition, the experimental rdf is much more favorable than the memoryless Gaussian rdf, and usually more favorable than the rdf for dependant sample Gauss-Markov data.

Any source encoder/decoder has some implied corresponding source model, or at least represents a reasonable compromise between obtaining the rdf and limiting the implementation complexity. In practice, stationary data models are usually used to design source encoders for the nonstationary image source. Adaptive encoders are often only subsequent modifications of stationary encoders, designed to overcome the observed effect of nonstationarity. The composite nonstationary source model model leads to a more fundamental treatment of adaptive image compression methods.

Universal coding uses the switched subsource model. In universal coding theory, one subsource is initially selected from the group of subsources, and the particular subsource selected is determined by observation of the output symbols. The subsource identification is used in encoding and decoding the source symbols, and is transmitted to the decoder using overhead rate. Universal coding theory shows that universal codes exist which achieve the rdf of any possible subsource. This is true because the overhead transmission is negligible, for long block length symbol codes. This method and result are included in figure 1 and equation 3 above. Gray and Davisson [5] observe that this result applies to nonstationary sources, if the switch position changes slowly. Gray and Davisson elsewhere [6] give an example of noiseless coding for image data, where the average transmission rate is less than the mixture entropy.

In the switched subsource model, the nonstationarity is limited to intermediate time intervals. For "short" time intervals, usually only one stationary subsource is switch selected, and the nonstationary source is "locally stationary". For "long" time intervals, the stationary switch statistics produce the symbol statistics of the mixture source, and the nonstationary source is "long term stationary". If one of the subsources of the composite source is nonstationary, it may also be modeled as a composite source, and then combined with the original composite source so that only one switch is used in the model. If the composite source switching is nonstationary, the source can be modeled as a group of composite subsources, each with different stationary switching statistics, one of which is selected by a second switch. This requires a multiplication of encoder/decoders, since the subsource probabilities affect the rate and distortion allocated to each subsource. Since the rate and distortion are average measures, the loss in modeling a low probability nonstationary subsource as stationary is relatively minor.

THE STATIONARY IMAGE MODEL AND EXPERIMENTAL IMAGE STATISTICS

The sampled image process is usually modeled as a wide-sense stationary, first-order Markov process.[7,8] The stationary source model and the mixture source statistics provide important information about the composite source model. For the samples x_i, $i = 1, 2, \ldots, N$, the mean, variance, covariance, and correlation coefficient are defined as follows:

$$E(x_i) = \mu$$

$$E\left[(x_i - \mu)^2\right] = \sigma^2$$

$$E\left[(x_i - \mu)(x_j - \mu)\right] = c_{ij}$$

$$r_{ij} = c_{ij} / \sigma^2$$

$$= \frac{E(x_i x_j) - \mu^2}{E(x_i^2) - \mu^2} \qquad (6)$$

By definition of the first-order Markov process,

$$r_{ij} = r^{|i-j|} \qquad (7)$$

The optimum source coding for the first-order Markov porcess is well known[1], and requires a simple predictor and difference encoder. The best estimate of x_i, given all the x_j, for $j < i$, is identical to the best estimate given x_{i-1}. The optimum predictive source coder is shown in figure 2. The optimum estimate of x_i is

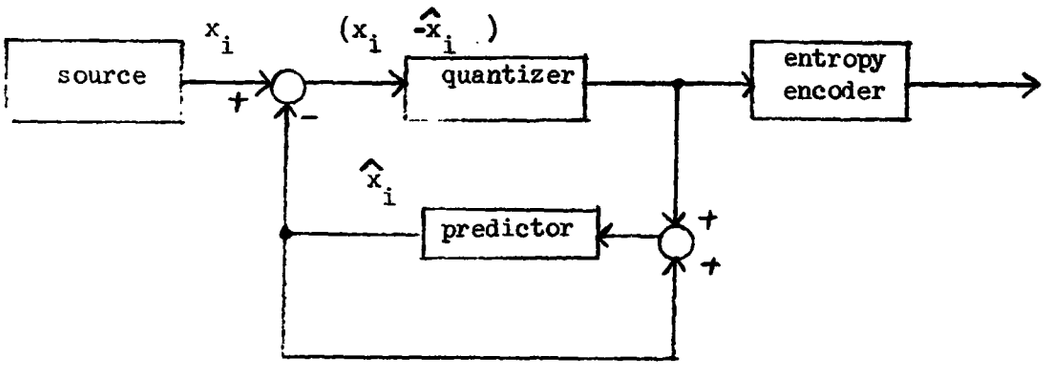


Figure 2. The optimum predictive source encoder for a first-order Markov process.

$$\hat{x}_i = r\,(x_{i-1}) + \mu$$

$$= r\,x_{i-1} + \mu(1-r) \qquad (8)$$

The predictor bias and mean-square error are

$$E(x_i - \hat{x}_i) = 0$$

$$E\left[(x_i - \hat{x}_i)^2\right] = \sigma^2\,(1-r^2) \qquad (9)$$

In source coding, both the encoder and decoder compute x_i, and the difference $(x_i - \hat{x}_i)$ is computed at the encoder, quantized, and transmitted to the decoder. The rate distortion function of the first order Markov source with Gaussian sample distribution is [1]

$$R(D) = \frac{1}{2}\,\log_2\,\frac{\sigma^2(1-r^2)}{D} \qquad \text{for} \quad D \le \frac{1-r}{1+r} \qquad (10)$$

For r=0, the rdf is that of the memoryless Gaussian source, $R(D) = \frac{1}{2}\,\log_2\,(\sigma^2/D)$. σ^2 is the variance of the original source samples. The optimum encoder transmits the optimum predictor differences, which have variance $\sigma^2(1-r^2)$.

It is of interest to determine the statistics of the experimental image test set to be encoded. The statistical measurements corresponding to equation 6 are as follows:

$$\bar{X} = \frac{1}{2(N-d)} \sum_{i} (x_i + x_j)$$

$$S^2 = \frac{1}{2(N-d)} \sum_{i} (x_i - \bar{X})^2 + (x_j - \bar{X})^2$$

$$mc_{ij} = \frac{1}{N-d} \sum_{i} (x_i - \bar{X})(x_j - \bar{X})$$

$$mr_{ij} = \frac{mc_{ij}}{S^2} \qquad (11)$$

where $i = i, 2, \ldots, N-d$
and $j = i + d$

These equations define the experimental mean, variance, covariance, and correlation. The addition of two similar factors, and the compensating factor of 1/2, occur in $\bar{X}$ and S^2 so that each of the x_i are used the same number of times as in mc_{ij} and mr_{ij}. d is the sample distance for the covariance and correlation measurement.

These statistics were measured for the experimental test set of five images. The original samples are quantized to six bit accuracy, and the statistics of table I are in units of the least significant bit. Values of mr (d=1) reported in the literature are usually between 0.95 and 0.99, and this is true for all the test images except the highly detailed Band image. The dependance of correlation on sample distance corresponds to the first-order Markov model, as has been shown previously [9]. The values of $\bar{X}$ and S^2 reflect the average brightness and contrast of the image, and are not indicators of the image information content.

Table I. Experimental mean, variance and correlation for five experimental test images.

Image	Mean	Variance	Correlation
Reasoner	1.12	94.17	0.987
Two Girls	-10.92	91.48	0.980
Two Men	-8.81	172.62	0.973
Writing Pad	12.18	43.83	0.979
Band	-12.85	92.24	0.871
Average	-3.86	98.87	0.958

The optimum predictor and rdf of equations 8 and 10 are time invariant for a wide sense stationary process. If the video source were actually wide sense stationary, the statistics of equation 11 and table I would have the same expected values, independant of the particular image or region of an image measured. However, the video image source is well known to be nonstationary, so that using different encoder/decoders at different times gives improved performance. The interpretation of the full image statistics of table I depends on the subsource switching rates considered in the composite source model. If the switching occurs frequently during a single image, the full image statistics define different mixture sources. If the encoder/decoder is selected once per image, the full image statistics are examples of subsources.

The composite source model to be investigated, based on the stationary model of the image source described above, consists of a group of first-order, wide sense stationary sources, one of which is selected by a switch. The first-order Markov model is specified by μ, σ^2, and r. If the subsources are defined using all three of these parameters, a large number of subsources results. In the most accurate subsource definition, the values of μ, σ^2, and r could be periodically measured, quantized to some accuracy, and used to design the encoder/decoder. The transmission rate overhead would be prohibitive, unless the subsource switching is very slow. The effect of these three parameters on the rdf is examined, and experimental measurements are made, to determine the effectiveness of subsource definitions based on these parameters.

NONSTATIONARY CORRELATION AND ADAPTIVE PREDICTION

In this section, the subsources are defined using only the distance one correlation, r. The value of r directly affects the predictor, quantizer, and entropy encoder, as shown in equations 8 and 9. With the image sample variance, the value of r determines the rdf, as shown in equation 10. The measured values of r in table I have significant differences between images, and areas of low detail and high correlation, and high detail with low correlation, are found in many typical images, including all five experimental test images.

The importance of a source parameter can be determined by the effect of a source-encoder mismatch in that parameter. We consider the effect of an arbitrary predictor on the predictor bias and predictor mean-square error. The arbitrary predictor corresponding to equation 8 is

$$\hat{x}_i = a(x_{i-1} - b) + b$$

In general a does not equal r, and b does not equal μ. The bias and mean-square of the predictor are

$$E(x_i - \hat{x}_i) = E(x_i - a(x_{i-1} - b) - b)$$

$$= (1-a)(\mu - b)$$

$$E\left[(x_i - \hat{x}_i)^2\right] = E(x_i^2) - 2E(x_i \hat{x}_i) + E(\hat{x}_i^2)$$

$$= \sigma^2 (1 - 2ar + a^2) + (\mu - b)^2 (1-a)^2 \qquad (12)$$

The second term in the mean-square error [illegible] of the bias. These equations reduce to 9 for a = r, b [illegible]

Since the experimental correlation is usually close to one, it is reasonably effective to set a = 1, b = 0, in a non-adaptive image compression predictor. The predictor bias is zero, and the mean-square error is $2\sigma^2(1-r)$, where r and σ^2 are the actual image statistics. The minimum mean-square error is $\sigma^2(1-r^2)$, and is obtained when a = r, b = μ. The increase in error for a = 1, b = 0, is $\sigma^2(1-r)^2$, which is small when r is approximately one.

These results can be used to evaluate the effect of using the a = 1 predictor for subsources having r not equal to one. The correlation, error, and rate change gain based on the rdf are given in table II for a predictor using the correct value of r, and in table III for a predictor using a = 1. A larger negative value of rate change indicates that less rate is required, and higher compression is obtained. As expected, there is very little difference in performance of the two predictors for correlations approximately equal to one. The performance difference is also small for the lower correlations, because of the small potential compression gain of the optimum predictor.

We consider the effect of using a = 1, for two simple composite sources. A typical value for r is 0.95. If the source is stationary, the use of a = 1 rather than a = 0.95 increases the rate by 0.01 bits per sample. Suppose the source is nonstationary, having a subsource with r = 1.0, selected with 95 percent probability, and a subsource with r = 0.0, selected with 5 percent probability. Tables II and III show that the required rate is increased 0.5 bits per sample when r = 0.0. Since this occurs 5 percent of the time, the average rate is increased 0.025 bits per sample. If the subsources were r = 1.0 with 90 percent probability, and r = 0.5 with ten percent probability, the rate is increased 0.21 bits per sample, ten percent of the time, for an average of 0.021 bits per sample.

The potential additional compression gain of a nonstationary source model based on different values of correlation is very small. This explains Habibi's observation [3] that adaptive predictors for image data have not been reported in the open literature. Habibi characterizes the method of Tasto and Wintz [4,10] as an adaptive Karhunen-Loeve transform (KLT), but it has been shown that the KLT designed for the experimental correlation, like the predictor, is very little better than the KLT for correlation one.[11] (The KLT for r = 1 is identical to the discrete cosine transform.) The method of Tasto and Wintz uses a source partition based

Table II. Correlation, Predictor mean-square error, and rate change using the optimum predictor a = r, for the first order Markov model with $\mu = 0$, $\sigma^2 = 1$.

Correlation, r	Error, $\sigma^2(1-r^2)$	Rate Change, bits $0.5 \log_2$ Error
0.99	0.0199	-2.83
0.95	0.0975	-1.67
0.90	0.19	-1.20
0.80	0.36	-0.736
0.70	0.51	-0.500
0.50	0.75	-0.208
0.00	1.00	0.00

Table III. Correlation, predictor mean-square error, and rate change using the predictor a = 1.0, for the first order Markov model with $\mu = 0$, $\sigma^2 = 1$.

Correlation, r	Error, $2\sigma^2(1-r)$	Rate Change, bits $0.5 \log_2$ Error
0.99	0.020	-2.82
0.95	0.100	-1.66
0.90	0.20	-1.16
0.80	0.40	-0.662
0.70	0.60	-0.352
0.50	1.00	0.000
0.00	2.00	+0.500

on the experimental mean and variance, as well as the correlation. Even though the compression gain varies considerably as a function of correlation, little additional compression gain is obtained by adapting the encoder/decoder for nonstationary correlation. The typical average correlation is high, and the encoder for high correlation can be used at low correlation, since there is little potential gain to be lost.

This conclusion was tested by examining the performance of adaptive predictors on the five test images. A local predictor could be implemented by using the experimental mean and correlation of a local sample block, and transmitting the local mean and correlation with the predictor errors. It is desireable to avoid the additional overhead of transmitting the local mean. This can be done by setting b equal to the mean of the samples in several preceeding blocks, which requires no overhead. The local predictor is

$$\hat{x}_i = a(x_{i-1} - b) + b$$

The bias and predictor mean-square error are given in equation 12.

$$\begin{aligned} E\left[(x_i-\hat{x}_i)^2\right] &= E\left[(x_i-b-x_i+b)^2\right] \\ &= E\left[(x_i-b)^2\right] \\ &\quad - 2a\,E\left[(x_i-b)(x_{i-1}-b)\right] \\ &\quad + a^2\,E\left[(x_{i-1}-b)^2\right] \end{aligned} \qquad (13)$$

For minimum mean-square error, the derivative with respect to a is set to zero.

$$a = \frac{E\left[(x_i-b)(x_{i-1}-b)\right]}{E\left[(x_{i-1}-b)^2\right]}$$

Table IV. gives the predictor error and rate change for three different predictors used on the five test images. Results are given for a equal to one, and for a equal to the full image correlation of table I. The largest improvement using the full image correlation, 0.047 bits per sample, is obtained for the Band image. This indicates that changing the value of a slowly with respect to the video image rate can give a small average rate reduction. The predictor error and rate change are also given for local blocks of four samples. This predictor has smaller error and larger compression than the other predictors shown, but the overhead rate to describe the value of r (with accuracy of ±0.02) is about four bits per block, or one bit per sample. The overhead far exceeds the rate gain. The use of larger and smaller sample blocks similarly provides no net rate gain. The use of limited systems with two (r = 1.0, 0.0) or four (r = 1.25, 1.00, 0.75, 0.50) predictors reduces the subsource indication overhead, but also reduces the subsource identification gain. In several experiments, no overall rate gain was obtained.

The experimental results are in agreement with the theoretical results obtained with the typical full image statistics. Adaptive predictors are not useful for image data. This is not unexpected, since none have been reported in the image coding literature. The effect of nonstationary correlation on the quantizer and entropy encoder is considered below.

Table IV. Predictor error and rate change for five test images.

	Predictor						
	a = 1		a = full image r		a = local r		
	Error	rate ch.	Error	rate ch.	Error	rate ch.	overhead
Image							
Reasoner	.0271	-2.603	.0272	-2.600	.0180	-2.898	+0.882
Two Girls	.0340	-2.440	.0336	-2.448	.0233	-2.712	+0.984
Two Men	.0466	-2.211	.0461	-2.220	.0329	-2.462	+1.067
Writing Pad	.0304	-2.520	.0300	-2.529	.0186	-2.875	+0.898
Band	.2519	-0.995	.2359	-1.042	.1740	-1.262	+1.212

NONSTATIONARY PREDICTOR ERROR VARIANCE

In predictive compression, the difference between the predicted next sample value and the actual value is quantized, entropy coded, and transmitted (figure 2). The rate distortion function, given in equation 10, is a function of the predictor error variance, $\sigma^2 (1-r^2)$. The predictor error variance depends on both the original sample variance and on the sample correlation. When the predictor error variance is nonstationary, the optimum encoder/decoder has several subsource encoder/decoders with different quantizers and entropy coders. As in the previous section, the range and effect of the parameters used to define the subsource, $\sigma^2 (1-r^2)$, is estimated. In following sections, adaptive entropy encoders and adaptive quantizers are described.

The range of $\sigma^2 (1-r^2)$ depends on r and σ^2. The values of r (mr) measured using equation 11 on a large number of samples can range from 0 to 1, but are typically 0.9 to 1.0 for image data. The experimental image data is quantized to six bits, two's complement, and sample values range from -32 to +31. The sample values often have a uniform or peaked distribution, so that σ^2 can be estimated from the data range. For a uniform distribution with range $\pm A$, $\sigma^2 = A^2/3$. For A = 31, $\sigma^2 = 320$. For A = 16, $\sigma^2 = 85$. For a triangular distribution with range $\pm A$, $\sigma^2 = A^2/6$, one-half the uniform distribution value. For A= 31, $\sigma^2 = 160$. For A= 16, $\sigma^2 = 42$.

The values of r and σ^2 for the experimental test images are given in table I. r ranges from 0.87 to 0.99, with an average of 0.96. Three images have σ^2 of 93 ±2, and the other two have σ^2 = 44 and 173, approximately half and double the value for the other three images. The Writing Pad ($\sigma^2 = 44$) has large regions of white, and Two Men ($\sigma^2 = 173$) has considerable sharp background detail.

Equation 10 can be rewritten

$$R(D) = (1/2) \log_2 \sigma^2 + (1/2) \log_2 (1-r^2) - (1/2) \log_2 D \qquad (14)$$

The first two terms define the change in the rate distortion function due to changes in σ^2 and in r. The effect of r on R(D) is given in table II, and the effect of σ^2 on R(D) is easily computed.

We first consider a source model with subsources having the statistical variation typical of full images, as given in table I. Suppose the average of the subsources has sample variance σ^2, correlation r, and rdf R(D). The two subsources are defined as follows:

Subsource 1

$$\sigma_1^2 = 0.5\sigma^2, \quad r_1 = r, \quad R_1(D) = R(D) + (1/2)\log_2(1/2) = R(D) - 0.5$$

Subsource 2

$$\sigma_2^2 = 1.5\sigma^2, \quad r_2 = r, \quad R_2(D) = R(D) + (1/2)\log_2 1.5 = R(D) + 0.292$$

The two subsources have equal probability, so that the average of the subsource has variance σ^2. When a different encoder/decoder is used for each subsource, the subsource encoder achieves the average of the subsource rdf's.

$$R_{ss}(D) = (1/2) \quad R_1(D) + R_2(D)$$

$$= R(D) - 0.104$$

The gain of a system adapting on this model, which corresponds roughly to a different subsource for each image, is about 0.1 bit per sample. Although small, this gain is larger than the gain of an adaptive predictor.

We next consider the more widely differing subsources that might occur within an image. Suppose that subsource 1 consists of highly correlated, low detail or flat image regions comprising 75 percent of the image, and that subsource 2 consists of the remaining low correlation, high detail regions, such as edges. The average variance is σ^2, the average correlation is r = 0.95, and the rdf is R(D). The two subsources are defined as follows:

Subsource 1. (flat)

$$\sigma_1^2 = 0, \quad r_1 = 1.0, \quad R_1(D) = 0$$

Subsource 2. (edge)

$\sigma_2^2 = 4\sigma^2$, $r_2 = 0.8$, $R_2(D) = R(D) + (1/2)\log_2 4 + (1/2)\log_2 (1-r_2^2)/(1-r^2)$,

$R_2(D) = R(D) + 1.0 + 0.942 = R(D) + 1.942$

Using the appropriate encoder/decoder for each subsource, the subsource rate is

$$R_{ss}(D) = (3/4)\, R_1(D) + (1/4)\, R_2(D)$$

$$= (1/4)\, R(D) + 0.486$$

The relative gain depends on the stationary mixture source rate, R(D), and is given in table V. The table shows that substaintial rdf improvements are possible for the flat/edge model, but the assumption that three-quarters of the image can be transmitted at zero rate is extreme.

The next section describes adaptive entropy encoders, which change according to the predictor error distribution.

Table V. Rate gain for subsource coding of the flat/edge model.

$$R_{ss}(D) = (1/4)\, R(D) + 0.486$$

R(D)	$R_{ss}(D)$	change	percent
1.0	0.736	0.264	26.1%
2.0	0.986	1.014	50.7%
3.0	1.236	1.764	58.8%
4.0	1.486	2.514	62.9%

ADAPTIVE ENTROPY CODING

In this section, we consider noiseless coding. No quantizer is used, and the exact predictor difference value is transmitted without distortion. For noiseless coding, the rdf results of the previous sections can be simplified. The rdf is the minimum mutual information,

$$I(x;y) = H(x) - H(x/y) \tag{15}$$

where H(x) is the entropy of x, and H(x/y) is the conditional entropy of x given y. For distortionless transmission, H(x/y) = 0, and R(0) = H(x). The required rate is simply the entropy of x. If the source producing the output symbols is nonstationary, the conditional entropy is reduced by considering the memory or an equivalent subsource identification. The entropy can be measured directly, and Wyner and Ziv [12] have shown that the entropy reduction for a source with memory is a bound on the rdf reduction. Specifically,

$$R_{mix}(D) - R(D) \leq H_{mix}(x) - H(x) \tag{16}$$

The reduction in the rdf of a source due to memory is less than or equal to the reduction in source symbol entropy.

Table VI shows the entropy of the differences and of the difference magnitudes for the five test images. Comparing the difference entropy with the values of r and σ^2 given in table I shows that the general effect of these parameters on entropy and rdf agrees with the theory examined in the previous sections. For example, Writing Pad has correlation nearly identical to that of Two Girls, but has about one-half the sample variance, and the difference entropy is 0.89 bits less. Two Men also has similar correlation, but has nearly twice the sample variance of Two Girls, and has a difference entropy 0.59 bits greater. Reasoner and Band have sample variance very similar to Two Girls, but have higher and lower correlation and the difference entropy is correspondingly smaller and greater.

Table VI. Entropy of the differences and difference magnitudes for the five test images, in bits.

Image	Difference Entropy	Magnitude Entropy	Sign Entropy
Reasoner	2.05	1.62	0.43
Two Girls	2.72	2.06	0.66
Two Men	3.31	2.61	0.70
Writing Pad	1.83	1.43	0.40
Band	3.78	3.10	0.68
Average	2.74	2.16	0.58

However, if the average entropy is used as a reference point, computation of the change in entropy using equation 14 and the values of σ^2 and r from table I leads to some discrepancies, as shown in table VII. This is apparently due to the effect of varying experimental difference distributions. The rdf of equations 10 and 14 applies to the Gaussian difference distribution, with variance $\sigma^2(1-r^2)$. It has been shown [13] that the entropy and rdf for any symmetrical distribution are equal to the Gaussian entropy and rdf, plus a constant depending on the distribution. Therefore, if the test images had the same difference distribution except for the variance $\sigma^2(1-r^2)$, equations 10 and 14 would yield the correct differences in rdf and entropy. The exact differences are not obtained because the test image difference distributions have different shape as well as scale. This problem also affects quantizer design, considered in the next section.

The entropy of the difference magnitudes is also given in table VI, and is used below rather than the difference entropy. In the first order Markov data model, the sample values depend only on the previous sample, and the sample differences are uncorrelated. This implies that the signs of the predictor differences are independant. The first order model may not be strictly true for the experimental data, but using the difference magnitude entropy eliminates the effect of correlated signs. The magnitude entropy is the information contained in all bits except the sign bit, and table VI shows that the sign bit actually contains only about one-half bit of information. The measured average probability of the difference being equal to zero is 0.42, for the five test images. The sign bit has one bit of information, but is used only 58 percent of the time, so its average information content is 0.58 bits, in agreement with the average sign entropy of table VI.

In a first experiment in adaptive entropy coding, we consider the performance of a single fixed entropy coder used on all five test images. The combined difference entropy of the five images, based on the combined difference probabilities, is 2.966 bits. A simple approximate Huffman code is shown in figure 3, and its rate performance is given in table VIII. The average rate achieved for the five test images is equal to their combined entropy, within the computational accuracy. Table VIII also shows that using the optimum entropy code for each image provides an average rate reduction of 0.23 bits. This is larger than the estimate of 0.1 bits for full image nonstationarity, made in the previous section.

Table VII. Computed differences in entropy for the five test images.

Image	Change in difference entropy from the average of table VI.	Change predicted using the r^2 of table I.
Reasoner	-0.69	-0.87
Two Girls	-0.02	-0.58
Two Men	+0.57	+0.09
Writing Pad	-0.91	-1.08
Band	+1.04	+0.73

Figure 3. Approximate Huffman code.

Difference Magnitude	Code Word	Length
0	0	1
1	10S	3
2,3	110SM_1	5
4,5,6,7	1110SM_1M_2	7
8,9,..., 15	11110S$M_1M_2M_3$	9
16,17, ...,31	111110S$M_1M_2M_3M_4$	11
32,33, ...,63	111111S$M_1M_2M_3M_4M_5$	12

S indicates the sign bit, and M_i indicates the ith magnitude bit.

Table VIII. Performance of the approximate Huffman code of figure 3.

Image	Difference Entropy	Rate for Approximate Huffman code	Rate Increase
Reasoner	2.05	2.19	0.14
Two Girls	2.72	2.96	0.24
Two Men	3.31	3.64	0.33
Writing Pad	1.83	2.01	0.18
Band	3.78	4.04	0.26
Average	2.74	2.97	0.23

The observed nonstationarity within the test images, and the computation of potential rate gain in the last section, indicate that further rate gain can be obtained by adapting the entropy coding within the image. Local nonstationarity appears as nonstationary predictor error variance, which causes the sample difference magnitudes to be dependant. The dependance may be significant over a span of many samples, although the closest are usually more similar.

The potential rate reduction due to local nonstationarity can be bounded by the entropy reduction for the difference magnitude, when the difference magnitude is conditioned on the previous difference magnitude. The conditional magnitude entropy can be found by computing the entropy of pairs of differences.

$$H(|x_j - \hat{x}_j| \,/\, |x_i - \hat{x}_i|) = H(|x_j - \hat{x}_j| , |x_i - \hat{x}_i|) - H(|x_i - \hat{x}_i|)$$

The conditional magnitude entropy is shown in table IX for the five test images, and is equal to the rate required to transmit noiselessly the current difference magnitude, given the previous difference magnitude. The average rate gain is only 0.17 bits per sample, or 7.8 percent of the magnitude entropy. The gain is high for Reasoner and Band, which have a few large areas of similar samples, and is low for Two Girls, which has small detail throughout the image.

Although the subsource definition and the current difference value are mutually dependant, some functional distinction between subsource and difference information must be made. Here, the subsource definition will include information having significant dependancies between sucessive sample differences. The subsource identification is largely determined by the most significant magnitude bits of the closer differences, and has little relation to the sign bits or less significant bits.

We first consider the case where the subsource identification is based only on the magnitude of the current difference, but uses the difference correlation. That is, the subsource identification is the non-independant part of the difference

Table IX. Reduction in entropy for conditional difference magnitudes.

Image	$H(\lvert x_j-\hat{x}_j\rvert, \lvert x_i-\hat{x}_i\rvert)$ / $H(\lvert x_i-\hat{x}_i\rvert)$	$H(\lvert x_j-\hat{x}_j\rvert / \lvert x_i-\hat{x}_i\rvert)$	Gain
Reasoner	3.03	1.62	1.41	0.21
Two Girls	4.04	2.06	1.98	0.08
Two Men	5.08	2.61	2.47	0.14
Writing Pad	2.71	1.43	1.28	0.15
Band	5.92	3.10	2.82	0.28
Average		2.16	1.99	0.17

information. The subsource identification contains some of the difference data, and must be transmitted for each difference. Since the same information is transmitted for independant differences, there is no overhead penalty when there is no difference correlation.

This system was implemented experimentally by dividing the difference data by powers of two, that is, 1, 2, 4, etc., and treating the quotient (the most significant bits) as the subsource identification, and the remainder as data. The sum of the subsource and data entropy is equal to the original difference magnitude entropy, but the subsource identification entropy is reduced when the subsource identification is conditioned on the previous subsource identification. This data is shown in the second numerical column of table X, for division by 2. There is one data bit (with entropy very nearly 1.0) and the remaining difference magnitude information defines the subsource. As expected, this method has performance similar to transmission of the difference magnitude conditioned on the previous difference magnitude. For division by 4 or 8 there is little information in the subsource identification, since most differences are small.

In another experiment, the subsource identification is set equal to the location of the most significant bit in the difference magnitude. There are seven subsources, numbered 0 through 6 as shown.

7 difference bits	7	6	5	4	3	2	1	
value	±	32	16	8	4	2	1	
7 subsources		6	5	4	3	2	1	0
magnitude range		31	15	7	3	1	0	0

Given the subsource identification (the location of the largest non-zero magnitude bit), the range of the remaining magnitude data is limited. The subsource identification is conditioned on the previous subsource, and the remaining data is entropy coded, conditioned on the subsource. The total entropy results of this subsource identification are shown in the third numerical column of table X. This and the previous method are similar to conditional encoding of the difference

Table X. Total magnitude entropy for several methods of subsource identification.

	Subsource Identification				
	Magnitude entropy conditioned on previous magnitude	Difference magnitude divided by 2	Location of most significant magnitude bit	Largest magnitude in block of 16	Standard deviation of block of 16
Image					
Reasoner	1.41	1.44	1.42	1.42	1.44
Two Girls	1.98	1.99	2.00	2.03	2.02
Two Men	2.47	2.55	2.48	2.44	2.44
Writing Pad	1.28	1.33	1.31	1.35	1.38
Band	2.82	2.88	2.84	2.90	2.92
Average	1.99	2.04	2.01	2.03	2.04

magnitudes, but are slightly worse in performance and are simpler in implementation.

It is more usual, especially in adaptive quantization, to define the subsource using some measured statistic of a block of difference magnitudes. Experiments wera also performed using the subsource defined as the largest difference magnitude in a block, and as the standard deviation of the differences in a block. For these two methods, and the five test images, the minimum entropy usually occured for sample difference blocks of length 8 or 16, with one minimum at 32. The data is given for both methods, for blocks of 16, in table X. Performance is similar to that of the other two methods described above. A further rate reduction of about 0.02 bits can be made if the subsource identification is conditioned on the previous subsource identification, as for the other two methods.

A comparison of the methods of adaptive entropy coding considered in this section is given in table XI. The results are all given in terms of difference entropy, by adding the sign entropy of each image (table VI) to the magnitude entropies. The result is correct if the difference signs are mutually independant, and independant of the difference magnitude, as in the first order Markov data model. The simplest method, with the highest rate, is to use the simple fixed approximate Huffman coder of figure 3 for all images. Using the optimum coder for each image gives a 7 percent rate reduction. Using the local nonstationarity or subsource identification information provided by the the previous difference or neighboring differences gives an additional 6 percent rate reduction.

The small gains of adaptive entropy coding are not untypical, considering results reported in the literature. Rice and Plaunt [14] used an adaptive variable length noiseless coding system for image data. The most appropriate code for the the sample differences was selected for each block of 21 samples. The system produced rates within 0.25 bits of the entropy for test image areas with a wide range of entropy. Rates below the average entropy were not sought or observed. Spencer and May [15] compared Rice and Plaunt's technique to the method of using the previous line statistics to generate a code for the current line. Both mehtods gave a 10 percent gain in rate over the optimum full image Huffman code. Davisson and Grey [6] combined a run length code, three variable length codes, and direct difference transmission, and selected the best method for blocks of 64 samples.

Table XI. Comparison of adaptive entropy coding methods.

Image	Fixed approximate Huffman code	Sample difference entropy	Previous difference conditional entropy	Subsource identification (best method)
Reasoner	2.19	2.05	1.84	1.85
Two Girls	2.96	2.72	2.64	2.66
Two Men	3.64	3.31	3.17	3.18
Writing Pad	2.01	1.83	1.68	1.71
Band	4.04	3.78	3.50	3.52
Average	2.97	2.74	2.57	2.59
Percent change	0	-7.3%	-13.5%	-12.8%

They noiselessly compressed an image with 3.3 bits per sample average entropy to 3.0 bits per sample.

Although the gain of adaptive entropy coding for the nonstationary image model is only 0.4 bits, this gain is an order of magnitude greater than the gain of adaptive prediction. The entropy gain is an upper bound on the rdf improvement due to nonstationarity. For the five test images, the rate improvement due to nonstationarity is limited to 0.4 bits, when we go from a simgle fixed method for all five images to a locally adaptive method. The rate improvement is limited to 0.17 bits, on the average, when we go from the optimum fixed method for a single image to a locally adaptive method. In the next section, we consider adaptive quantizers, which achieve higher rate gains, for reasons not related to nonstationarity.

ADAPTIVE QUANTIZATION

The previous section considered entropy or noiseless coders used without quantizers. This section considers quantizers, both with and without entropy coders. Quantizers reduce the transmission rate, at the cost of reduced fidelity. Memoryless quantizers operating independantly on the predictor errors can perform close to the rdf for stationary data.[16] The optimum quantizers have been found for several predictor error distributions, including the Gaussian, exponential, and gamma distributions.[17,18] The theoretically optimum quantizers are scaled according to the standard deviation of the quantized variable. The Gaussian quantizers typically have one-half the range of the exponential or gamma quantizers, and the latter are more suited to image predictor difference data.

The most efficient quantization method is uniform quantization followed by entropy coding.[19] Performance is within 1/4 bit per sample of the Gaussian rdf.[16] The optimum non-uniform quantizer, without an entropy coder, requires a 20 percent plus 1/8 bit rate increase over the Gaussian rdf.[16] Addition of entropy coding to the optimum non-uniform quantizers brings performance close to that of the optimum uniform quantizers with entropy coding. The performance penalty for the optimum non-uniform quantizers without entropy coding is larger for the exponential and gamma distributions than for the Gaussian distribution.[20]

Quantizers introduce both in-range and out-of-range distortion. If the quantizer range is less than the full possible difference range, large differences are represented by smaller difference values. This causes some large errors in the reconstructed samples, but the probability of large differences is small. Such errors are visible in images as edge blurring or slope overload. If the smallest quantizer interval size is larger than the least significant bit of the original data, the reconstructed samples will have small random errors, similar to the errors produced by original quantization using too few bits. These errors are visible as contouring in the flat, low contrast areas of the image. Quantizer designs are optimized by considering these two kinds of errors. If a uniform quantizer is gradually widened from the minimum range, the total mean-square error is first decreased as slope overload is reduced, and then increased as contouring becomes

pronounced. Optimum non-uniform quantizers obtain the best possible performance by using larger quantization intervals at the larger, less probable difference values.

The first step in investigating the performance of adaptive quantizers is to test fixed quantizers for the five test images. The experimental data are six bits and sample values range from -32 to + 31. The difference data are integers, with a possible range of -63 to +63. The difference data for these images, and for images in general, usually have an exponential distribution, and a scaled discrete version of some theoretical quantizer could be used. Instead, the experimental data were used to design the quantizers. A computer program was written to find the minimum mean-sqare error discrete quantizer, by exhaustive search. The quantizers were designed using the difference magnitude probabilities, and therefore are symmetrical about the zero difference point. To minimize the mean-square error, differences were assigned to the closest representative value.[23] For equally distant representative values, the smaller was used. Quantizers were designed for M, the number of representative values, equal to 2, 3, 4, 5, and 7.

For symmetrical distributions, quantizers with M odd always have zero as a representative value, while theoretical quantizers with M even usually have all the representative values symmetrical in pairs about zero. For image sample differences, the probability that the difference is zero is large, and quantizers with M odd usually have better performance than quantizers with a one-larger number of representaive values. This can be shown by a simple example. Suppose that the probability that the difference is zero is slightly larger than one-half, and the probability that the difference is ±1 is slightly less than one-half.

$$p(0) = 0.5 + e$$

$$p(\pm 1) = 0.5 - e$$

If M is equal to 2, the optimum symmetrical representative values are ±1, and the mean-square error is

$$MSE\ (M=2, \pm 1) = (0.5+e)\ 1^2 + (0.5-e)\ 0$$

$$= 0.5 + e$$

If all the differences are represented by zero, the error is smaller.

$$MSE\ (M=1, 0) = (0.5+e)\ 0 + (0.5-e)\ 1^2$$

$$= 0.5 - e$$

Because of the superior performance of M odd quantizers, the optimum quantizers for experimental difference data for M even usually have only M - 1 levels, with a zero representative value. M even quantizer designs were obtained by not allowing zero to be a representative value.

If all the sample differences are represented by zero, no difference data is actually transmitted, and the image or line is represented by the first sample. Obviously it is much better to transmit a ±1 indication of the sample change (delta modulation) than to send no information. The above anomalous result occured because the effect of the quantizer is analized as if the quantizer followed the differencing loop of figure 2, instead of being within the loop. In such an analysis, the quantizer is designed for the original sample differences, not for the differences between the current sample and the previous transmitted sample, which has been reconstructed from a sequance of quantized differences. Treating the quantizer as outside the loop gives acceptable results only when the quantizer error is small. Table XII shows for the Reasoner image, the estimated mean-square error of the optimum quantizer computed as if the quantizer were outside the loop and the actual measured mean-square error when the quantizer is used in the loop. For large M, the estimated error is nearly correct, but for M equal to 2, the estimated error is an order of magnitude smaller than the measured error.

The above result indicates that it is unlikely that low rate, high distortion quantizers designed for the original sample differences will give the optimum

Table XII. The optimum quantizers, computed for the Reasoner image sample differences, and the estimated and measured mean-square error.

Number of levels M	Quantizer representative levels	Estimated error	Measured error
15	0,1,2,3,4,7,10,13	0.0229	0.0225
7	0,1,4,9	0.1549	0.2771
5	0,1,6	0.397	1.266
4	1,6	0.963	1.714
3	0,5	0.946	3.807
2	1	2.023	23.979

performance in actual use. This observation led to the use of an iterative, interactive program to obtain the best fixed quantizers for the five test images. This computer program iteratively designs the optimum quantizers for the actual difference distributions produced by the previously tested quantizers. Operator interaction is provided to terminate repetitive searches and to input additional test quantizers. The best quantizer results produced by this method are given in table XIII, for the five test images. Except for M equal to 3, performance for the Reasoner image is improved.

The effect of full image nonstationarity can be estimated from the results shown in table XIV. The same quantizations (the optimum quantizations for the Reasoner image) were used for all five test images. Although these are not the optimum quantizations for the group of five test images, the results provide an upper bound on the error of the optimum quantizers. The increase in mean-square error over that for the best fixed quantizers for each image ranges from 12 to 70 percent. This corresponds to an average transmission rate increase of about 0.3 bits per sample. Using the optimum quantizers for Band for all five test images gave higher average mean-sqaure error.

As we did for adaptive entropy coding, we next consider the effect of local nonstationarity within the image. Four different methods of subsource identification were used for adaptive entropy coding, and all four were found nearly equal in performance. For adaptive quantization, we test only one approach, described by Ready and Spencer.[21] A block of sample differences is quantized using several different quantizers, and the resulting mean-square error is computed. The transmission consists of the identification of the quantizer with the smallest error, and the corresponding quantized data. The best mean-square error results were obtained using four quantizers, and blocks of four sample differences. These results are shown in table XV.

The experimental rate and mean-square error distortion are plotted for Reasoner in figure 4. Several compression methods are included, and similar results are given for the Band image and the average of the five images in figures 5 and 6.

Table XIII. The best fixed quantizers, and resulting mean-square error, found by iterative search.

	Number of representative levels				
	M= 2	M = 3	M = 4	M = 5	M = 7
Image					
Reasoner					
levels	4	0,5	1,8	0,1,8	0,1,4,11
error	8.83	3.81	1.19	0.76	0.214
Two Girls					
levels	3	0,3	1,4	0,1,6	0,1,4,9
error	4.61	2.62	1.11	0.79	0.321
Two Men					
levels	4	0,7	2,9	0,3,10	0,1,4,11
error	12.05	4.68	2.58	1.45	0.851
Writing Pad					
levels	2	0,3	1,4	0,1,4	0,1,4,9
error	2.52	1.05	0.69	0.24	0.105
Band					
levels	5	0,9	2,11	0,5,14	0,3,8,17
error	19.89	9.36	5.15	3.48	1.17
Average					
error	9.58	3.40	2.14	1.34	0.64

Table XIV. Mean-square error using the same quantizations for all five test images. (The Reasoner quantizations of table XIII are used.)

	Number of representative levels				
	M = 2	M = 3	M = 4	M = 5	M = 7
Image					
Reasoner	8.83	3.81	1.19	0.76	0.214
Two Girls	6.14	2.17	1.58	1.23	0.370
Two Men	12.05	5.55	2.85	2.58	0.851
Writing Pad	5.61	2.14	0.89	0.43	0.113
Band	20.95	13.04	6.60	6.37	2.87
Average	10.72	5.34	2.62	2.27	0.88

Table XV. Mean-square error for adaptive quantization.

Transmission rate	1.5	2.1	2.5
Quantizers	1 3 6 9	0,1 0,3 0,7 0,11	0,1 1,6 2,10 3,15
Image			
Reasoner	1.58	0.39	0.30
Two Girls	1.72	0.56	0.56
Two Men	3.70	1.37	1.05
Writing Pad	0.97	0.21	0.19
Band	9.48	4.16	2.25
Average	3.49	1.34	0.87

FIGURE 5. RATE DISTORTION FOR THE BAND IMAGE

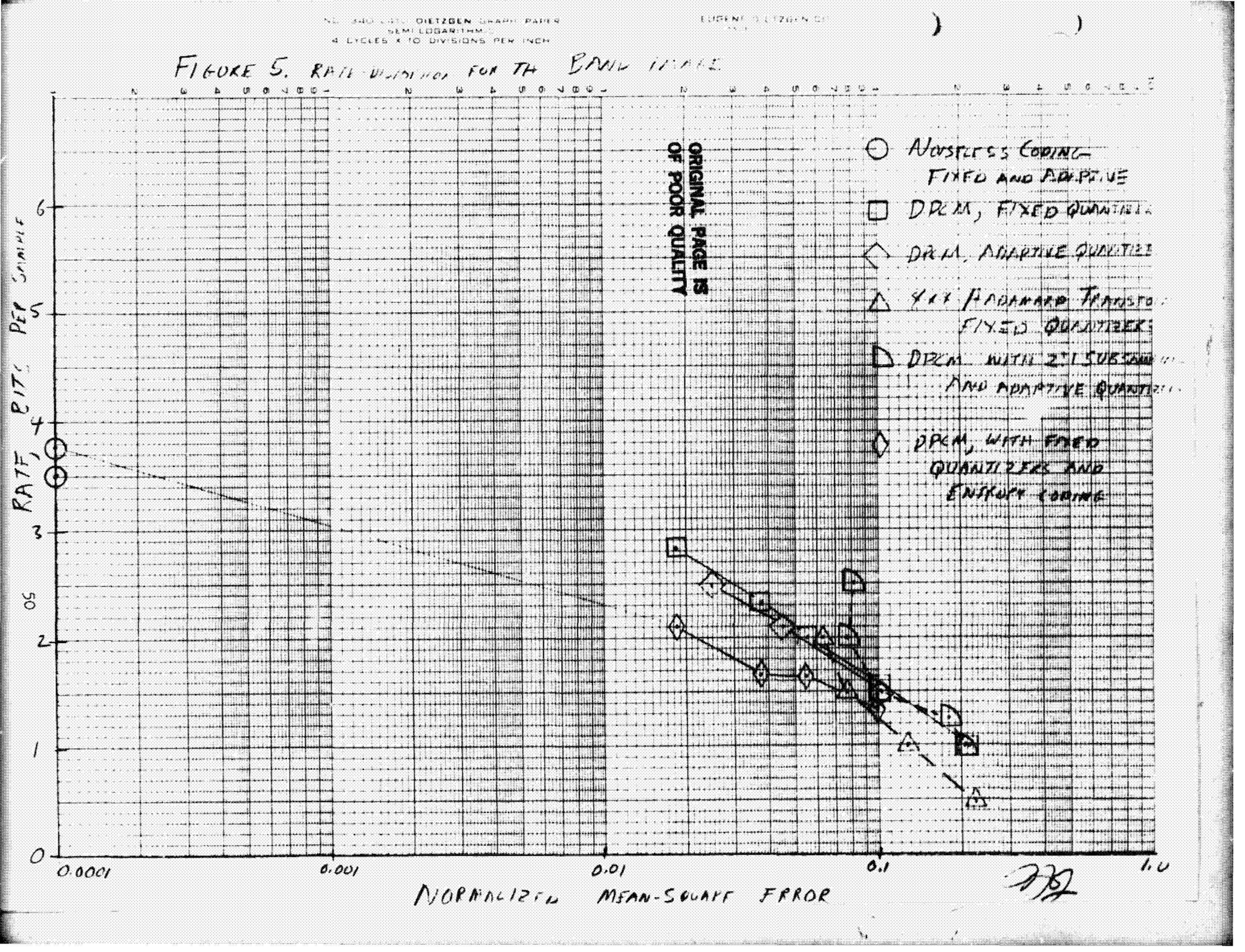


FIGURE 4. RATE - DISTORTION FOR THE REASONER IMAGE

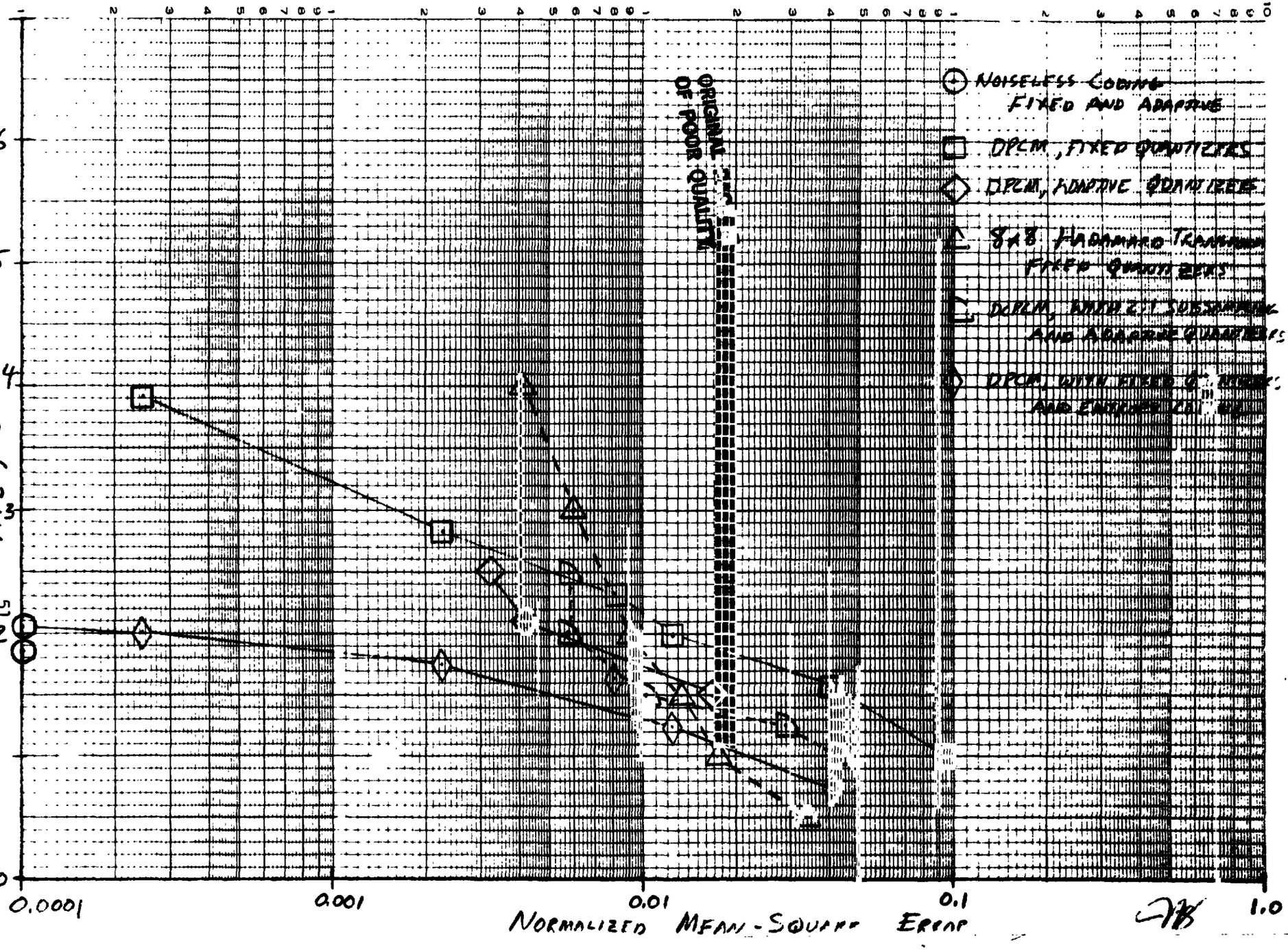
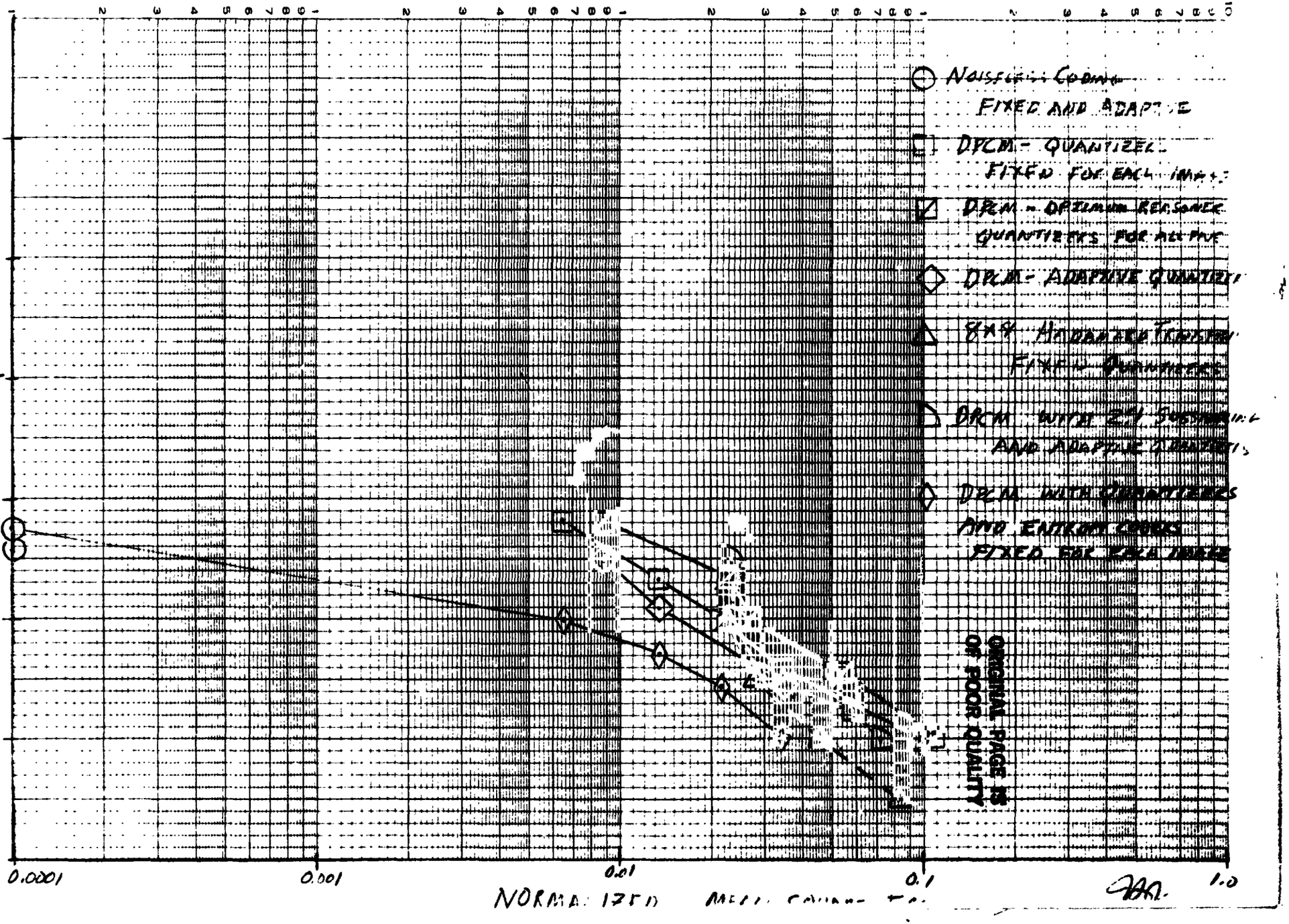


FIGURE 6: RATE-DISTORTION FOR THE FIVE IMAGE AVERAGE



Adaptive quantization gains about 0.5 bits per sample for the Reasoner image, compared to the best fixed quantizers, but adaptive quantization gains only about 0.1 bits per sample for Band, and 0.1 to 0.2 bits per sample for the five image average. The high entropy, high distortion Band image tends to dominate the five image average measurements. Using one method of adaptive quantization for all the test images gives better performance than using the optimum fixed quantizer for each image, and is more easily implemented. The gains due to between-image and local nonstationarity are very similar to those of adaptive entropy coding, where coders designed for each image gained 0.25 bits per sample over a fixed code for each image, and a locally adaptive entropy coder gained a further 0.17 bits per sample.

The adaptive quantization rate gains for the four test images not including Band are about twice as large as the rate reduction due to adaptive entropy coding. Since the reduction in entropy is an upper bound on the reduction in the rdf, some of the gain of adaptive quantization is due to reducing the inefficiency of fixed quantizers.

Figures 4, 5, and 6 also show the rate-distortion results of two dimensional Hadamard transform compression, performed on 8 by 8 blocks of samples. The compression method is fixed for each rate, not adaptive, and has been described previously.[22,23] The results shown are for independant field compression, rather than interlaced frame compression. Frame compression requires more memory, but produces higher compression gain. The transform coefficient quantizers were designed to reflect the observed flat-edge nonstationarity of the test images. The comparison of the results in figures 4, 5, and 6 illustrates the well-known fact that predictive coding is much better than transform coding at high rates (3-4 bits per sample), and much worse than transform coding at low rates (1 bit per sample).[24,25,26] In the Reasoner data shown in figure 4, a mean-square error less than about 0.01 corresponds to an acceptable compressed image. Transform coding has relatively high, although subjectively unobjectionable, mean-square error at high rates because of the large number of computations and coefficients affecting each reconstructed sample. Predictive coding with a = 1 is mathematically simple. Transform coding is more sucessful than predictive coding at low rates for several reasons. The

8 by 8 transform block is two dimensional, and includes a larger number of horizontally and vertically correlated samples. At the lowest rates, the best transform compression designs effectively perform horizontal and vertical subsampling.

For predictive coding, the effect of 2 to 1 horizontal subsampling was tested by combining it with adaptive quantization. The results are shown in table XVI, and are also plotted in figures 4, 5, and 6. As expected, subsampling provides improved performance at lower rates, while introducing some fixed error at the higher rates. Those samples omitted by subsampling, are reconstructed by averaging the adjacent reconstructed samples. As a minor benefit, subsampling allows the overall rate to be determined more flexibly, by allowing quantizers with larger numbers of levels to be used at lower total rates.

Predictive coding, adaptive quantizer predictive coding, or adaptive quantizer predictive coding with subsampling are able to perform better than transform coding at high rates down to 2 or 1.5 bits per sample. Although transform coding is much better than the predictive methods at lower rates, even the best transform images at such lower rates are subjectively unacceptable.

Table XVII shows the entropy of the best fixed quantizers of table XII, and also the noiseless entropy form table VI. These results have also been plotted in figures 4, 5, and 6. This data shows that the predictive methods using quantization with variable rate entropy coding are significantly superior in performance to the fixed quantizer methods. Two dimensional transform is superior at the lowest rates.

The rate distortion curves for figures 4, 5, and 6 can be approximated by a straight line drawn between the zero rate, 100 percent error rate point at the lower right corners, and the noiseless coding rate points on the left axes. The plotted results of entropy coding the output levels of the optimum quantizers are all within 0.5 bits per sample of the rdf, and are much closer at the lower rates. Entropy coding obtains similar significant rate gains for theoretical distributions, and a further slight improvement can be obtained by entropy coding the output levels of the optimum uniform quantizers. [16,19,20]

Table XVI. Mean-square error for 2 to 1 horizontal subsampling with adaptive quantization.

Transmission rate	1.0	1.25	1.5	2.0	2.5
Image					
Reasoner	4.056	2.713	0.972	0.545	0.546
Two Girls	2.926	2.202	1.369	0.951	0.835
Two Men	8.322	6.276	3.315	2.416	2.387
Writing Pad	1.188	0.953	0.495	0.372	0.314
Band	19.106	16.703	9.220	7.241	7.455
Average	7.119	5.768	3.074	2.306	2.307

Table XVII. The quantizer output entropy for the best fixed quantizers of table XIII.

	Number of representative levels					
	M = 2	M = 3	M = 4	M = 5	M = 7	noiseless
Image						
Reasoner	1.0	0.728	1.235	1.638	1.751	2.05
Two Girls	1.0	1.323	1.627	1.949	2.134	2.72
Two Men	1.0	1.069	1.389	1.691	2.404	2.31
Writing Pad	1.0	0.874	1.317	1.604	1.610	1.83
Band	1.0	1.136	1.630	1.653	2.098	3.78
Average	1.0	1.026	1.440	1.707	1.999	2.74
Maximum entropy (equally likely)	1.0	1.585	2.000	2.322	2.807	

CONCLUSION

The improved performance of adaptive compression techniques is well known. The major assumption of this investigation was that the gains of adaptive compression are due to the nonstationarity of the image source. This is commonly accepted. [3,21,27] The objective was to use the nonstationary source model to obtain the maximum additional data compression due to nonstationarity. Contrary to the above assumption, it was found that the rate distortion bound improvement due to image nonstationarity is small, and that significant performance gains in compression systems can not be made by considering the source nonstationarity. The gains of adaptive quantizer coding are due more to the inefficiency of non-adaptive fixed rate quantizers in approaching the rdf, than to the source nonstationarity.

In the experiments leading to this conclusion, the mean-square error results of several different compression methods based on one dimensional predictive coding, and one method of two dimensional transform compression were compared. The relative performance is in agreement with results in the literature, in that predictive coding is superior to transform at higher rates and inferior at lower rates, and in that variable rate entropy coding gives performance near the rate distortion bound.

REFERENCES

1. T. Berger, Rate Distortion Theory, Englewood Cliffs, N.J., Prentice-Hall, 1971.

2. R. Ash, Information Theory, New York, Interscience, 1965.

3. A. Habibi, "Survey of Adaptive Image Coding Techniques," IEEE Transactions on Communications, vol. COM-25, Nov. 1977, pp. 1275-1284.

4. M. Tasto and P. A. Wintz, "A Bound on the Rate-Distortion Function and Application to Images," IEEE Transactions on Information Theory, vol. IT-18, Jan. 1972, pp. 150-159.

5. R. M. Gray and L. D. Davisson, " A Mathematical Theory of Data Compression?", Proc. 1974 International Conference on Communications, pp. 40A-1,40A-5.

6. L. Davisson and R. Gray, "Advances in Data Compression," in Advances in Communications Systems, vol. 4, A. V. Balakrishnan and A. J. Viterbi, editors, New York, Academic Press, 1975, pp. 199-228.

7. L. E. Franks, "A Model for the Random Video Process," Bell System Technical Journal, April, 1966, pp. 609-630.

8. E. R. Kretzmer, "Statistics of Television Signals," Bell System Technical Journal, July, 1952, pp. 751-763.

9. H. W. Jones, "A Comparison of Theoretical and Experimental Video Compression Designs, IEEE Transactions on Electromagnetic Compatibility, Feb. 1979,

10. M. Tasto and P. A. Wintz, "Image Coding by Adaptive Block Quantization," IEEE Transactions on Communication Technology, vol. COM-19, Dec. 1971, pp. 957-972.

11. H. W. Jones, D. N. Hein, and S. C. Knauer, "The Karhunen-Loeve, Discrete Cosine, and Related Transforms Obtained Via the Hadamard Transform, 1978 International Telemetering Conference, vol. 14, Nov. 1978, pp. 87-98.

12. A. D. Wyner and J. Ziv, "Bounds on the Rate-Distortion Function for Stationary Sources With Memory," IEEE Transactions on Information Theory, vol. IT-17, Sept. 1971, pp. 508-513.

13. H. W. Jones, "Theoretical and Subjective Bit Assignments in Transform Picture Processing," NASA Tech. Rep. TN D-8440, Mar. 1977.

14. R. F. Rice and J. R. Plaunt, "Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data," IEEE Transactions on Communications Technology, vol. COM-19, Dec. 1071, pp. 889-897.

15. C. L. May and D. J. Spencer, "ERTS Image Data Compression Technique Evaluation," Final Report for NASA Contract NAS5-21746, April 1974. Ref. 43 in Habibi, reference 3 above.

16. T. J. Goeblick and J. L. Holsinger, "Analog Source Digitization: A Comparison of Theory and Practice," IEEE Transactions on Information Theory, vol. IT-13, April 1967, pp. 323-326.

17. J. Max, "Quantizing for Minimum Distortion," IRE Transactions on Information Theory, vol. IT-6, March 1960, pp. 7-12.

18. M. D. Paez and T. H. Glisson, "Minimum Mean-Squared Error Quantization in Speech PCM and DPCM Systems," IEEE Transactions on Communications, vol. COM-20, April 1972, pp. 225-230.

19. R. C. Wood, "On Optimum Quantization, " IEEE Transactions on Information Theory, vol. IT-15, Mar. 1969, pp. 248-252.

20. H. W. Jones, "Minimum Distortion Quantizers," NASA Tech. Rep. TN D-8384, Mar. 1977.

21. P. J. Ready and D. J. Spencer, " Block Adaptive DPCM Transmission of Images," NTC'75 Conference Record, vol. 2, pp 22-10 to 22-17.

22. H. W. Jones, A Comparison of Theoretical and Experimental Video Compression Designs," IEEE Transactions on ElectroMagnetic Compatibility, vol. EMC-21, Feb. 1979, pp. 50-56.

23. H. W. Jones and L. B. Hofman, "Comparison of Video Fields and Frames for Transform Compression," SPIE Vol. 149 Applications of Digital Image Processing, Aug. 1978, pp. 214-221.

24. A. Habibi, "Hybrid Coding of Pictoral Data, IEEE Transactions on Communications, vol. COM-22, May 1974, pp. 614-624.

25. A. Habibi, Comparison of nth-Order DPCM Encoder with Linear Transformations and Block Quar. ization Techniques, IEEE Transactions on Communication Technology, vol. COM -19, Dec. 1971, pp. 948-956.

26. J. B. O'Neal and T. R. Natarajan, "Coding Isotropic Images," IEEE Transactions on Information Theory, vol. IT-23, Nov. 1977, pp. 697-707.

27. A. G. Tescher and R. V. Cox, "An Adaptive Transform Coding Algorithm," ICC'76, International Conference on Communciations Conference Record, June 1976, pp. 47-10 to 47-23.

SECTION D

SEL 32 COMPUTER PROGRAMS

AND VIDEO DATA

This report describes the computer programs and compressed video data developed on the SEL 32 computer under contract NAS2-9703. Previous work was reported in the interim final report, Section B, Appendix D. The interim final report indicated that nearly all of the original data four frame sets, monochrome sequences, and color sequences had been transferred to the SEL 32.

All computer programs are in Fortran, but they use assembly language routines found in the Video Library. The programs described here, and preliminary versions and minor variations, are all files under USERNAME JONES or USERNAME IMAGE. The programs have also been saved by username on two tapes. Because of the ready availability of these files, no listings are attached.

Table I indicates the six major functions of video processing that have been performed in the NASA-Ames Video Research Lab. The individual programs are listed under these functions.

1. Video Data Record

(none) - records monochrome and color video on digital tape
- currently unavailable due hardware modifications
- see old DREC and EREC, format information below

2. Test Video Display

JRAMP -Tests the display link, using a partial frame ramp on frame 1
- ACTIVATE JRAMP

KCOL - tests the display link, using horizontal color bars and letters
- ACTIVATE KCOL

3. Video Display

- all programs display both D and E format, color or monochrome

DISP 5,6 - displays all frames on a tape up to a double EOF
- 5 for M71C, 6 for M711
- ACTIVATE DISPLAY5 ACTIVATE DISPLAY6

JDISP - displays frames from selected tape onto selected frame
- requires both tape mounts and TYXX [1]
- ACTIVATE JDISP

Table I. SEL 32 Program Functions

1. Video Data Record

2. Test Video Display

3. Video Display

4. Video Digital Tape Reformat or Copy

5. Video Compression

   Transform Programs

   Differencing Programs

6. Video Data Analysis

(Video Display continued)

(none) - reformat tape for DICOMED
- not available

4. Video Digital Tape Reformat or Copy

DORE - converts D to E, E to D formats
- ACTIVATE DTOE, ACTIVATE ETOD
- to catalog, USE DORE, CH/DORE//DTOE/, RUN

ECOPY - copies a D or E tape, with video display
- catalog into JONESMOD

ERTOY[2] - converts YIQ to RGB, RGB to YIQ
- option to select frames
- requires TY60[1], discfile[3]
- catalog into JONESMOD

DFMT - converts DINT (frame) to DSTD (field), DSTD to DINT
- requires TY60, discfile
- catalog into JONESMOD

5. Video Compression

Transform Programs

E8X8 - 8x8 Hadamard transform, monochrome and color
- input E tape M710, output E tape M711
- requires TY60, discfile [1,3]
- uses CR78 cards to define bit assignments and quantizations[4]
- catalog into JONESMOD

T8X8 - like E8X8, includes cosine and quasi-cosine transforms
- additional cards to define post Hadamard vector transform matrix
- catalog into JONESMOD

E88F - 8x8 Hadamard conditional replenishment
- use E88FA for multiple tape input
- like E8X8
- additional cards for refresh lists, change threshold, mode threshold
- catalog into JONESMOD

(Video Compression, transform, continued)

FRAVG - averages E frames in time
- for output of E88F, ERTOY
- requires TY60, discfile [1,3]
- requires tape input, tape and/or display output

Differencing Programs

DCOR8 - adaptive prediction
- D tape input on M710
- requires TY64, discfile [1,3]
- catalog into JONESMOD

DCORC - prediction and adaptive entropy coding
- like DCOR8

DCOMP - prediction and best theoretical quantizers
- D input on M710, D output on M711
- requires TY60, discfile [1,3]
- catalog into JONESMOD

DCOMI - iterative, interactive quantizer selection
- like DCOMP

DCOMK - adaptively selected fixed quantizers
- like DCOMP

DCOML - subsampling, prediction, adaptive quantization
- like DCOMP

6. Video Data Analysis

DCOR - measures in-line and between line correlation
- like DCOR8

TQUAN - design optimum quantizers, as in DCOMP
- place in workfile, RUN
- requires TYXX

DDIF - measures MSE between D tapes
- not debugged

Notes:

1 TYXX assignments may be changed by recatalogging

2 Files above ERTOY are in USERNAME IMAGE, ERTOY and below are in USERNAMEJONES

3 Discfiles must be created before program use. Use the job outline in subroutine WRDISC to create the discfiles. WRDISC has many versions, so use the one in the program to be run.

4 The cards to run the transform compression programs are in the card file drawers.

We next consider the video tape formats. These have been transferred basically unchanged from the SEL 840, but the data is now in eight bit format rather than six bit format. The video data tape conversion programs, discussed in the interim final report, expanded the data bits but retained the original formats.

Video Tape Format Definition

The video data digital tapes consist of 1) title records, 2) data records, and 3) end of file marks (EOF's).

1. Title Records

Each video tape file has one standard title record of 25 words. A computer word is 32 bits.

First word - tape identification (A4)

Words 2 to 10 - miscellaneous alphanumeric and integer information, including date, etc. (15A4)

Words 11 to 25 - descriptive title (15A4)

Word 25 - format identification, DSTD, etc. (A4)

2. Data Records

Each video data digital tape has many data records based on the D format video line. A D format line has 107 words of 32 bits each.

First word - number of data words (always 104), an integer

Second word - line number (6 to 237, or slightly higher), an integer

Third word - video field or color number (0 to 7) an integer

Words 4 to 107 - Two's complement data, packed eight bits per picture element, four elements per word

A D format tape record consists of one D format line. An E format record consists of eight D format video lines, joined in sequence.

3. End of file marks (EOF's)

Each video tape file is marked by an EOF. Each tape has a second EOF after the last file.

A file may contain one, three, or four video frames. The first monochrome data gathered to simulate 4 by 4 by 4 Hadamard transform operation, has four monochrome frames in time sequance in each file. Test images taken from these, and longer sequences made using the video disc analog recorder, have only one monochrome frame in each file. Color files always contain three frames, one for each primary color.

4. Video Line Order

DSTD/ESTD. For both D and E format tapes, the standard line order is in the order of the temporal video scan. The lines are ordered 6 to 237 in field 0, then 6 to 237 in field 1, and so on to field 5 or 7 if there are three or four frames in the file.

(Video Line Order continued)

DINT/EINT. For both D and E format tapes, the interlaced format is organized by frames, rather than fields. Each line in the first or even field is followed by the corresponding (same numbered) line in the second or odd field. Relatively few data files are in the interlaced format, but conversion is simple, and the compression programs accept interlaced input.

5. Original Video Data Tapes

Many monochrome files (D101-125, D151-170) consist of eight fields in time sequence. Other monochrome files are a single frame (two fields) forming part of a sequence of about 85 frames (ET01, ET03, ET06, ET07).

Color files are either RGB or YIQ in frames 1,2, and 3, respectively. Two color sequences of 59 (EC09,10,11) and 46 (EC12,13) frames exist. The first has been reformatted in reverse field order. Single frames of higher color quality have also been obtained. They were recorded in the original D color format, which had a line given in each color sucessively, instead of by frames in each color. The files DC13-22 have been converted from old DINT, RGB to new line order EINT, YIQ using DFMT , DTOE, and ERTOY. DC01-10 could be similarly converted for use with current compression programs.

6. Compressed Video Data Tapes

Several demonstration tapes were prepared for a review in mid-January. These are described here by tape label.

T1 - DC13-22, EINT, processed by T8X8
- frame interlace, 8x8, cosine, color
- 2 bpp, 1.5 Y, 0.25 I and Q

C1 - Man and Tool, 85 frames, E88F
- 1/8 bpp, monochrome, Hadamard, 8x8

C2 - Wheel, 59 frames, E88F
- 1/4 bpp, color Hadamard, 8x8

A1 to A8 - Wheel, 234 frames with reversal and two runs, E88FA
- 1/2 bpp, color Hadamard, 8x8
- new 2.5 bpp Y edge mode, new edge threshold

DEMO22 - Reasoner, monochrome
- results of 21 waveform compression tapes
- original, Hadamard 3 and 1.5 bpp, DPCM 3 and 1.6 bpp
- adaptive DPCM and subsampling 1.5 bpp

SECTION E

CONFERENCE PAPERS

Two conference papers have been presented since the interim final report. The topics were "Comparison of Video Fields and Frames for Transform Compression," by Harry W. Jones and Larry B. Hofman, and "The Karhunen-Loeve, Discrete Cosine, and related Transforms Obtained Via the Hadamard Transform," by H. W. Jones, D. N. Hein, and S. C. Knauer. Original drafts of these papers were included in the interim final report, as appendices A and B of Section B. Copies of the papers from the proceedings are included here.

COMPARISON OF VIDEO FIELDS AND FRAMES FOR TRANSFORM COMPRESSION*

Harry W. Jones, Jr.
COM-CODE, Inc.
Mountain View, California 94043

and

Larry B. Hofman
Ames Research Center, NASA
Moffett Field, California 94035

Abstract

Because of the interlaced television scan, the two fields that form an interlaced video frame are generated 1/60 of a second apart. If the two fields are compressed independently, the correlation between adjacent lines is unused. The transmission rate can be reduced by using a field memory to form an interlaced frame. Four test images were processed as fields and as interlaced frames, using both theoretical and experimental compression designs. For comparable mean-square error and subjective appearance, field compression requires about one-half bit per sample more than frame compression. However, the overall transmission rate — the number of bits per image time the number of images per second — is more meaningful than the number of bits per sample. When transform compression at low transmission rates merges the adjacent lines, frame compression becomes similar to field repeating, and the memory can be reduced.

Introduction

This paper describes the results achieved, and the hardware required, for video compression using either fields or interlaced frames. Interlacing the video fields, and the inverse operation, requires substantial digital memory, but achieves a given compressed image quality using a lower transmission rate.

In television transmission, the scene is repeatedly scanned to form a field image of about 256 lines. As shown in Figure 1, each field consists of every second line in the full frame, and the alternating fields (represented by solid or dashed lines) are displaced vertically by one line. Two successive fields form the full video frame of about 512 lines. Fields are transmitted at the rate of 60 times per second, to avoid the objectionable flicker effect which occurs at lower rates, even though 30 or fewer images per second are sufficient for motion representation.(1)

It is possible to transmit sampled images at reduced bit rates because much of the information in samples taken at the Nyquist rate is redundant. The successive samples are not independent, and the video process can be described by a first order Markov model.(2) This Markov model fits the measured correlation of the four test images used here, as shown in Figure 2 for the image of newscaster Harry Reasoner, and in Table 1 for all four test images. The image frames usually have the highest correlation between adjacent samples in adjacent lines in a frame (different fields), the next highest correlation between adjacent samples in the same line, and the lowest correlation is between corresponding samples in the closest lines in a field (separated by an alternate field line). This is as expected, from the 4-to-3 width-to-height aspect ratio of the video frame, and because the four test frames have low motion or change between fields.

Because higher correlation allows more transmission rate compression, the correlation values indicate that it is most effective to use samples in the adjacent lines of alternate fields, next most effective to use samples in the current line, and least effective to use samples in the closest lines of the same field. The cost of using these samples is the memory required to store them. Using samples in the same line requires a few samples to a line of samples to be stored; using the closest lines in the same field requires several lines of memory; and using samples in the adjacent lines of the alternate field requires a full field of memory. Obtaining the lowest possible transmission rate can require much more memory than less efficient systems. This effect is also apparent in conditional replenishment systems, which use the correlation between successive frames in time.(3) (4)

*The research work leading to this paper was performed under contract NAS2-9703, sponsored by the National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California 94035.

Transform Compression Systems

Computer simulations of video image compression systems were undertaken to compare the performance of field and frame compression. All experiment involved single image compression of monochrome television images. Digitized images were obtained by sampling a standard NTSC baseband signal at 8.064×10^6 samples per second. Each sample was represented by a six-bit integer. The visible area of the images has 416 samples per line and 464 lines per frame. The nominal 512 samples and 512 lines includes samples in the horizontal retrace and lines in the vertical interval. The television images were compressed both as fields of 232 lines and as interlaced frames of 464 lines.

The compression experiments used Hadamard transforms of 8 by 8 subpictures. The coefficients of the 64 Hadamard vectors are used to represent the subpictures. The 8 by 8 Hadamard basis vectors are shown in sequency order in Figure 3. Sequency is defined here as the total number of white-black and black-white transitions, in the horizontal or vertical directions. If the Hadamard transform is not normalized, the vector coefficients have a possible range of 12 bits, since each is the result of 64 additions or subtractions of 6 bit numbers. The vector coefficients were first rounded to the 8 most significant bits, and then quantized to an 8-bit integer. Transmission rate compression is achieved by using fewer than 64 quantizer levels, and indicating each using a code word shorter than 6 bits. In the final compressed picture, each sample is represented using a 6-bit integer, as in the original image.

Figure 4 shows the hardware organization of the independent field transform compressor. The input lines are converted to digital samples, stored in an 8-line memory, transformed in 8 sample by 8 line blocks, and quantized. The quantized bit stream is transmitted, and the inverse process is used to generate analog video. The field compressor uses the correlation between the samples in a line and between the lines in a field. Some correlations are not used, since each 8 by 8 subpicture is processed independently.

Figure 5 shows an interlaced frame transform compressor, which performs the same functions as the field processor. It differs because the 8 by 8 subpictures are taken from an interlaced frame, rather than from one field. The 8 by 8 frame subpictures have one-half the height of field subpictures. In order to interlace a frame, the first field is held in memory until the second field is being generated. The fields are then interlaced and the subpictures are transformed. Information on the two fields is partly transmitted and partly stored during the second field time, and the stored information is transmitted during the next field time. The receiver output display is not synchronized to the data transmission, as it was in the field compressor. To provide the correct display, the receiver requires a compressed memory to hold the frame in the transmitted form. This memory is decoded twice, to provide the two fields. The memory required is one field at 8 bits and one-half of a compressed frame (assumed to require 1 bit per sample) at the encoder, and one compressed frame at the decoder (assumed to require 2 bits per sample), or the equivalent of one frame at 7 bits (i.e., 7 bit frames). This is the cost of using the correlation between adjacent lines in a frame rather than the correlation between the closest lines in a field.

The interlaced frame compressor of Figure 5 uses transmitter memories which are alternately fully written, and then fully read once, so that the memory information is of no further use during one-half of the memory cycle. A reviewer[5] of this paper has devised a frame interlace system which improves memory efficiency by reading four lines from memory, and then immediately replacing them with four current lines for future use. This is made possible by redefining the concept of an interlaced frame; rather than using a frame of two successive fields, the frame definition is changed every four lines. For the current input field, the first four lines are combined with four lines from the previous field, the second four lines are placed in memory for combination with the following field, and so on alternately every four lines. This improved design requires one-half frame of uncompressed memory at both the transmitter and receiver, for a total memory requirement of 4 bit frames.

System Simulation Results

Figure 6 shows the mean-square error results obtained (in units of the least significant of the six original bits) when the Harry Reasoner test image was compressed using theoretical compression designs. The different compression designs consist of the bit assignments and quantizers for each of the 64 Hadamard vector coefficients. The theoretical designs assumed the first order Markov correlation model (with the "assumed in design" values of Table 1), an exponential distribution for the transform vector coefficients, and the mean-square error measure. At the same rates, field compression produces larger error than frame compression, or, equivalently, field compression requires more rate for a given error. However, there are two cases in the field data, and one case in the frame data, where 1/2 or 1 bit per sample increases in the transmission rate produce

little or no reduction in error. The theoretical designs obviously do not make the best possible use of the transmission rate.

Figure 7 shows the mean-square error obtained when the Harry Reasoner test image was compressed using experimental compression designs. The curves are smooth, and added rate always reduces error. The experimental designs give much lower error than the theoretical designs. The field compression curve for the experimental designs (Fig. 7) is nearly identical to the frame curve for the theoretical designs (Fig. 6), from 4 bits per sample down to 1 bit per sample. The experimental designs used are similar to designs obtained by trial and error, but were generated using a formalized procedure based on the requirement of good representation for both the edges and the low detail areas in video images. For a full discussion of the theoretical and experimental designs used, see Ref. 6.

Figure 7 shows that, over most of the range of transmission rates, field compression requires a transmission rate about 50% greater than frame compression, for the same mean-square error. At the highest rate shown, 4 bits per sample, the mean-square error is caused by rounding all the transform vectors to 8 bits, and all methods give about the same error.

Figure 8 shows the mean-square error obtained using the experimental compression designs on all four test images, in both frame and field compression. Because of the wide range in the detail and correlation of the test images, the mean-square error at each transmission rate ranges over an order of magnitude, and the mean-square error is plotted on a log scale. Even though the test images differ greatly, the parallel curves of Figure 8 show that the increased rate required for field compression is nearly constant, about 1/2 bit per sample, for these images over the range of transmission rates between 1/2 and 2 bits per sample. It seems that the frame or field compression trade-off can be summarized as 4-7 bit frames of memory for 1/2 bit per sample in transmission rate.

The subjective impressions of the compressed images agree in quality ranking with the mean-square error results. Figure 9 shows the original image of Harry Reasoner. Figure 10 shows this image compressed using 1 bit per sample in the frame, and Figure 11 shows it compressed using 2 bits per sample in the field. The compressed images exhibit edge degradation, especially at the shoulders, lips, collar, and tie. The field image at 2 bits per sample has somewhat higher quality than the frame image at 1 bit per sample, as indicated by the error values of Figures 7 and 8. (Originals of all the test images are shown in Ref. 6.)

Time Effects

The above comparison of frame and field video compression considered only the quality of the individual images, and ignored the effects of motion and the time sequence of images. The two fields in a frame are generated 1/60 of a second apart in time, and motion tends to make the correlation between adjacent lines in a frame lower than the correlation between the closest lines in the same field. A fifth test image, of a blurred hand moving rapidly over a writing pad, was compressed in the same way as the four other test images. Because of the motion, the mean-square error was lower for field compression. (An original of the pad image is shown in Ref. 7.) Transform compression, especially at lower transmission rates, tends to average adjacent samples and lines. Two fields processed as a frame become similar, and high motion areas where the original fields differ become blurred. In frame compression of high motion scenes at low transmission rates, the motion update rate is the frame rate, 30 per second, rather than the field rate, 60 per second. Because the frame rate is adequate for representing motion, this is not an impairment.

These observations suggest a third approach to video compression. Since frame processing tends to average the two fields at lower transmission rates (which would reduce the vertical resolution and the motion update rate to one-half their original values), frame compression is more similar to field repeat compression than to independent, two-field processing. In field-repeat compression, only one-half the fields are transformed and transmitted, and each transmitted field is displayed twice at the decoder. Figure 12 shows the block diagram of a field-repeat compression system. As the field to be transmitted is generated, half the current information is transmitted and half is stored. The full compressed field is retained in the decoder, for repeated display. The total memory requirement for field-repeat compression is 1 1/2 bit frames, rather than 4 or 7 bit frames for frame compression. A real time hardware system using the Hadamard transform and field repeat has been previously described.[8]

Since a single field has only one-half of the samples in a frame, the same overall transmission rate is obtained when the number of bits per transmitted sample is doubled for field repeat. The overall rate is the number of bits per image multiplied by the number of images per second. Field repeat compression transmits only one-half the field images used in frame or field compression, as discussed above. The previous mean-square error results

also indicate the performance of field-repeat, since the same error in each transmitted field is obtained if one or two independent fields are transmitted. Field-repeat compression at 2 bits per sample has the same error as field compression at 2 bits per sample, but the overall transmission rate for field-repeat corresponds to that for frame or field compression at 1 bit per sample. Field-repeat compression can be compared to frame or field compression in Figures 6-8 by moving each point on the field compression curve to a point having the same mean-square error and one-half the transmission rate. This shows that the error is slightly lower for field-repeat compression than for frame compression, and much lower than for field compression.

Figure 13 shows a field-repeat image of the first field of Figure 11. This field-repeat image requires the same overall transmission rate as the frame processed image in Figure 10, having one field at 2 bits per sample rather than one frame at 1 bit per sample, and the subjective quality is similar. The field-repeat image has lower quality than the field compressed image, but that image has two independent fields at 2 bits per sample and requires twice the overall transmission rate. In field-repeat, vertical resolution is noticeably reduced in detailed areas, and quantization noise and contouring are more apparent in background areas. It should be reemphasized that field-repeat compression is appropriate at the lower transmission rates, where it is not possible to provide the full potential resolution of uncompressed video.

Conclusion

Experimental simulations of interlaced frame and independent field compression systems indicate that frame compression can achieve a transmission rate about 1/2 bit per sample lower than field processing at a given image quality, with the added requirement of 4 or 7 bit frames of memory. Frame transform compression can be used at lower transmission rates than field compression, but replaces the two fields in the frame with two similar combinations of the original fields. If it is decided to use only one field in field-repeat compression, performance similar to frame compression at low transmission rates can be obtained using only 1 1/2 bit frames of memory. A conditional replenishment compressor, which uses the correlation between successive frames, can be implemented using 7 bit frames of memory.(4)

Acknowledgment

The authors are grateful for valuable discussions with Scott C. Knauer of NASA Ames Research Center.

References

1. Brainard, R. C., Mounts, F. W., and Prasada, B., "Low-Resolution TV: Subjective Effects of Frame Repetition and Picture Replenishment," Bell System Technical Journal, pp. 261-271, Jan. 1967.

2. Franks, L. E., "A Model for the Random Video Process," Bell System Technical Journal, pp. 609-630, April 1966.

3. Haskell, B. G., "Interframe Coding of Monochrome Television - A Review," SPIE, Vol. 87, Advances in Image Transmission Techniques, pp. 212-221, 1976.

4. Jones, H. W., "A Conditional Replenishment Hadamard Video Compressor," SPIE Vol. 119, Applications of Digital Image Processing (IOCC), pp. 91-98, 1970.

5. Knauer, S. C., Ames Research Center, NASA, personal communication.

6. Jones, H. W., "A Comparison of Theoretical and Experimental Video Compression Designs," to be published in the IEEE Trans. on Electromagnetic Compatibility.

7. Jones, H. W., "A Real-Time Adaptive Hadamard Transform Video Compressor," SPIE Vol. 87, Advances in Image Transmission Techniques, pp. 2-9, 1976.

Table 1. Range of the correlation parameter R for D equal to 1 through 7, where $R = C^{1/D}$, C is measured correlation and D is sample distance

Picture	In-line	Between lines in field	Between lines in frame
Reasoner	0.966-0.987	0.967-0.973	0.984-0.989
Two girls	0.965-0.982	0.946-0.950	0.972-0.978
Two men	0.964-0.977	0.933-0.946	0.968-0.972
Band	0.847-0.882	0.807-0.877	0.888-0.916
Assumed in design	0.95	0.94	0.97

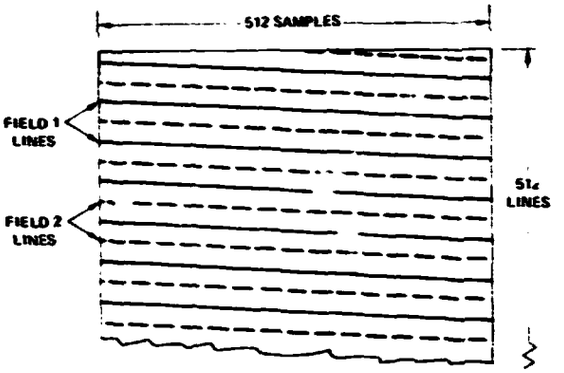


Fig. 1. The interlaced videoframe, with the first field given by solid lines and the second field given by dashed lines.

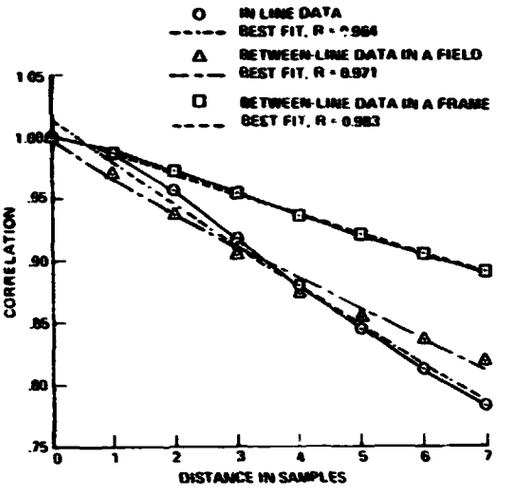


Fig. 2. Picture element correlation versus distance for the Reasoner image.

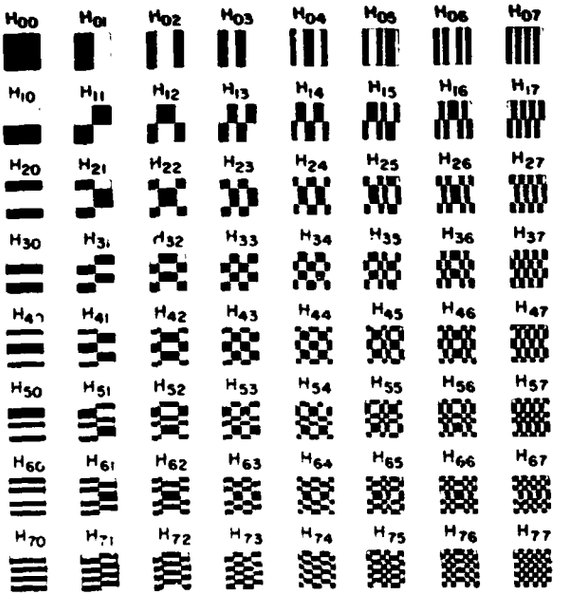


Fig. 3. The sequency ordered 8 by 8 Hadamard vectors.

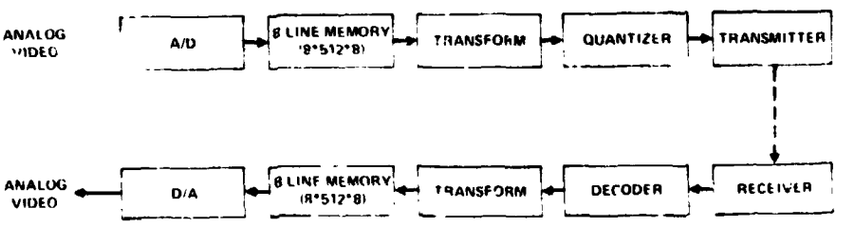


Fig. 4. Field transform compression.

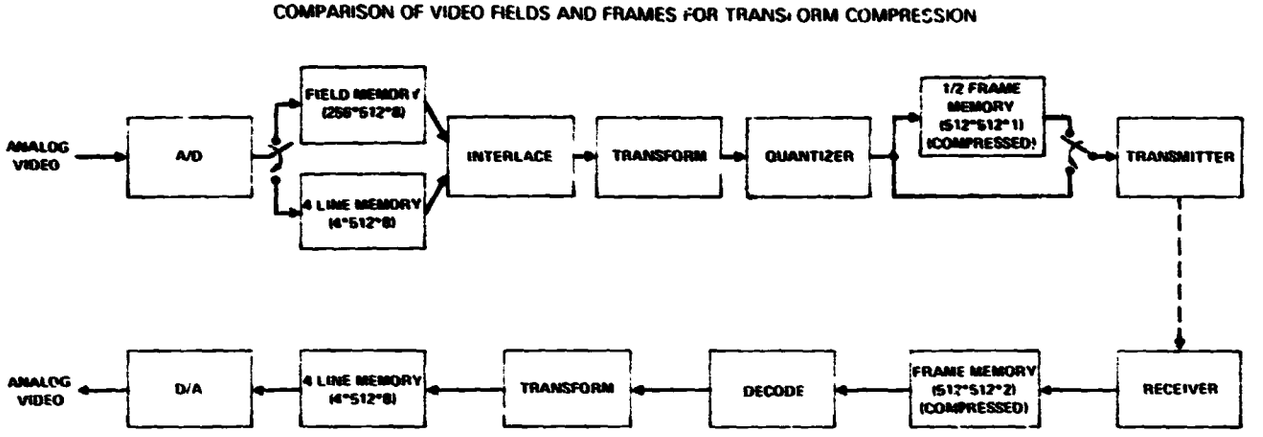


Fig. 5. Frame transform compression (7 bit frames of memory).

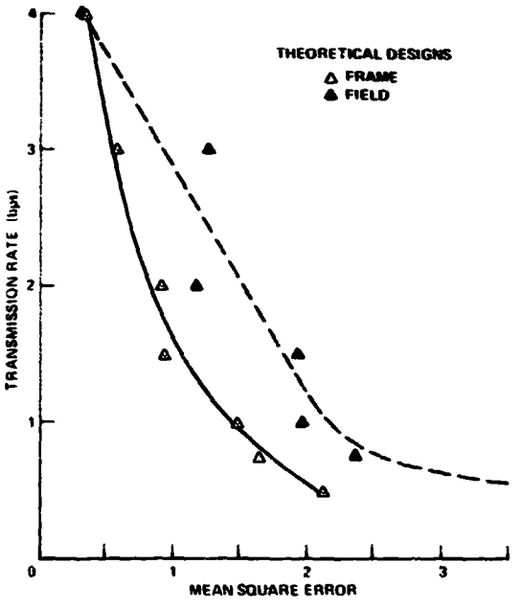


Fig. 6. Rate versus mean-square error for the Harry Reasoner image compressed using theoretical compression designs.

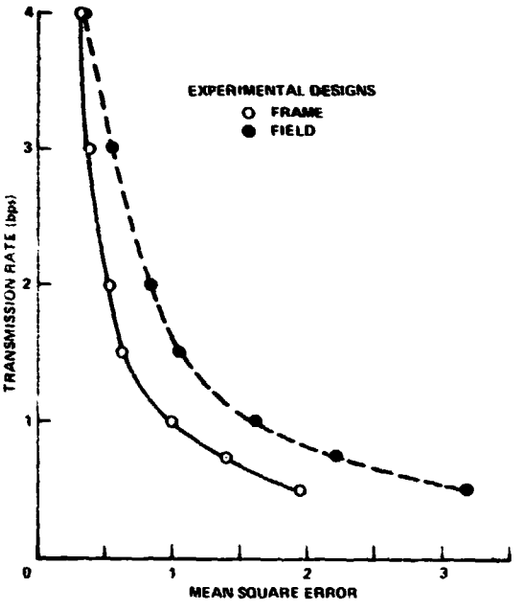


Fig. 7. Rate versus mean-square error for the Harry Reasoner image compressed using experimental compression designs.

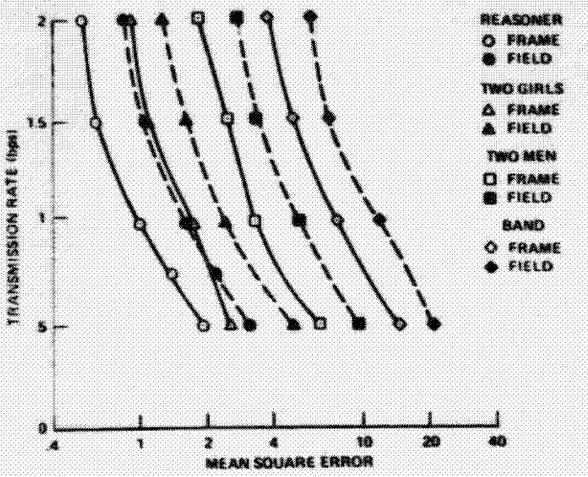


Fig. 8. Rate versus mean-square error for the four test images compressed using experimental compression designs.

Fig. 9. The original Reasoner image.

Fig. 10. The Reasoner image processed as an interlaced frame at 1 bpp, using the experimental design.

Fig. 11. The Reasoner image processed as two fields at 2 bpp, using the experimental design.

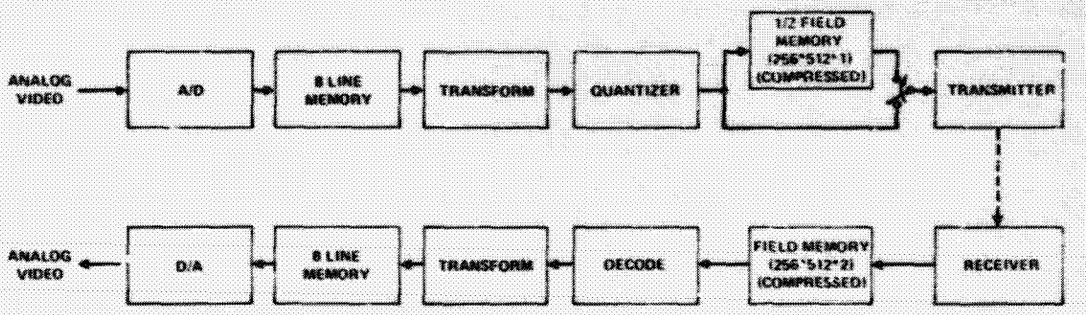


Fig. 12. Field repeat compression (1 1/2 bit frames of memory).

Fig. 13. The Reasoner image with the first field processed at 2 bpp using the experimental design (as in Fig. 11), and with the second field supplied by repeating the first field.

Proceedings of the Society of Photo-Optical Instrumentation Engineers

Volume 149

Applications of

Digital Image Processing

August 28-29, 1978, San Diego, California

Andrew G. Tescher
Editor

Cooperating Organization: American Meteorological Society

THE KARHUNEN-LOEVE, DISCRETE COSINE, AND RELATED TRANSFORMS OBTAINED VIA THE HADAMARD TRANSFORM*

H. W. Jones
COM-CODE, Inc.
305 Easy Street, No. 9
Mountain View, California 94043

D. N. Hein
Kansas State University
Manhattan, Kansas 66502

S. C. Knauer
Ames Research Center, NASA
Moffett Field, California 94035

ABSTRACT

The Karhunen-Loeve transform for stationary data, the discrete cosine transform, the Walsh-Hadamard transform, and most other commonly used transforms have one-half even and one-half odd transform vectors. Such even/odd transforms can be implemented by following a Walsh-Hadamard transform by a sparse matrix multiplication, as previously reported by Hein and Ahmed for the discrete cosine transform. The discrete cosine transform provides data compression nearly equal to that of the Karhunen-Loeve transform, for the first order Markov correlation model. The Walsh-Hadamard transform provides most of the potential data compression for this correlation model, but it always provides less data compression than the discrete cosine transform. Even/odd transforms can be designed to approach the performance of the Karhunen-Loeve or discrete cosine transform, while meeting various restrictions which can simplify hardware implementation. The performance of some even/odd transforms is compared theoretically and experimentally. About one-half of the performance difference between the Walsh-Hadamard and the discrete cosine transforms is obtained by simple post-processing of the Walsh-Hadamard transform coefficients.

INTRODUCTION

It is well known that the Karhunen-Loeve, or eigenvector transform (KLT), provides decorrelated vector coefficients with the maximum energy compaction, and that the discrete cosine transform (DCT) is a close approximation to the KLT for first-order Markov data (1). We will show that the general class of even/odd transforms includes this particular KLT, as well as the DCT, the Walsh-Hadamard transform (WHT), and other familiar transforms. The more complex even/odd transforms can be computed by combining a simpler even/odd transform with a sparse matrix multiplication. A theoretical performance measure is computed for some even/odd transforms, and two image compression experiments are reported.

EVEN/ODD TRANSFORMS

Orthogonal transforms are frequently used to compress correlated sampled data. Most commonly used transforms, including the Fourier, slant, DCT, and WHT have one-half even and one-half odd transform vectors. Several properties of such even/odd transforms are given in this section. The even vector coefficients are uncorrelated with the odd vector coefficients for a data correlation class which includes stationary data. The KLT is an even/odd transform for this class of data correlation. A conversion from one even/odd transform to another requires only multiplication by a sparse matrix, having one-half of its elements equal to zero.

If N, the number of data points, is an even number, a vector

$$V = (v_1 v_2 \cdots v_{N-1} v_N)^T$$

is said to be even if

$$v_i = v_{N+1-i} \qquad i = 1, \cdots, N/2$$

and is odd if

*Portions of this research were performed under NASA contract NAS2-9703 and NASA Interchange Agreement No. NCA2-OR-363-702.

$$v_i = -v_{N+1-i} \qquad i = 1, \cdots, N/2$$

For a data vector of length N

$$X = (x_1 x_2 \cdots x_{N-1} x_N)^T$$

the NxN correlation matrix is given by

$$\Sigma_x = E(XX^T)$$

Since Σ_x is a symmetric matrix, it can be partitioned into four N/2 x N/2 submatrices in the following manner:

$$\Sigma_x = \begin{vmatrix} A & B \\ B^T & C \end{vmatrix}$$

where

$$A = A^T \quad \text{and} \quad C = C^T$$

The general form for a transform matrix with one-half even and one-half odd basis vectors (called an even/odd transform), can be written as a partitioned matrix

$$H = \begin{vmatrix} E & \tilde{E} \\ D & -\tilde{D} \end{vmatrix}$$

where E and D are N/2 x N/2 orthogonal matrices, and $\tilde{E}$ and $\tilde{D}$ are formed by reversing the order of the columns in E and D, that is,

$$\tilde{E} = E\tilde{I} \quad \text{and} \quad \tilde{D} = D\tilde{I}$$

where the permutation matrix $\tilde{I}$ is the opposite diagonal identity matrix. The matrix H can then be factored into the product of two matrices

$$H = \begin{vmatrix} E & O \\ O & D \end{vmatrix} \begin{vmatrix} I & \tilde{I} \\ I & -\tilde{I} \end{vmatrix}$$

It is next shown that the even and odd vector coefficients of an even/odd transform are uncorrelated, for a general class of data correlation matrices. The correlation matrix for the transformed data vector, Y = HX, is given by the similarity transform.

$$\Sigma_y = H \Sigma_x H^T$$

$$= \begin{vmatrix} E(A + B\tilde{I} + \tilde{I}B^T + \tilde{I}C\tilde{I})E^T & E(A - B\tilde{I} + \tilde{I}B^T - \tilde{I}C\tilde{I})D^T \\ D(A + B\tilde{I} - \tilde{I}B^T - \tilde{I}C\tilde{I})E^T & D(A - B\tilde{I} - \tilde{I}B^T + \tilde{I}C\tilde{I})D^T \end{vmatrix}$$

The even and odd vector coefficients are uncorrelated when the opposite diagonal submatrices are identically zero. This is obviously true in the special case where

$$A = \tilde{I}C\tilde{I} \quad \text{and} \quad B = \tilde{I}B^T\tilde{I}$$

These equations state that the data correlation matrix Σ_x is symmetric about both the main diagonal and the opposite diagonal. This condition is satisfied if $E(x_i x_j)$ is a function of the magnitude of i - j, that is, if the process is stationary. For stationary data, the correlation matrix is a symmetric Toeplitz matrix (1, 2).

This decorrelation property of even/odd transforms is used to show that the KLT is an even/odd transform. For K, a reordered matrix of the KLT vectors, the transformed vector is Z, Z = KX. The correlation matrix for the transformed vector is given by

$$\Sigma_z = K \Sigma_x K^T$$

Suppose that the data are first transformed by the matrix H, above, and that the data are such that the even and odd coefficients are uncorrelated by this even/odd transform:

$$\Sigma_y = H \Sigma_x H^T$$

$$= 2 \begin{vmatrix} E(A + BI)E^T & 0 \\ 0 & D(A - BI)D^T \end{vmatrix}$$

$$= \begin{vmatrix} Y_1 & 0 \\ 0 & Y_2 \end{vmatrix}$$

Since H is invertible, K = AH, for $A = KH^T$.

$$\Sigma_z = AH \Sigma_x H^T A^T$$

$$\Sigma_z = A \Sigma_y A^T$$

$$\Sigma_z = A \begin{vmatrix} Y_1 & 0 \\ 0 & Y_2 \end{vmatrix} A^T$$

Suppose that

$$A = \begin{vmatrix} A_1 & A_2 \\ A_3 & A_4 \end{vmatrix}$$

$$\sum_z = \begin{vmatrix} A_1 Y_1 A_1^T + A_2 Y_2 A_2^T & A_1 Y_1 A_3^T + A_2 Y_2 A_4^T \\ A_3 Y_1 A_1^T + A_4 Y_2 A_2^T & A_3 Y_1 A_3^T + A_4 Y_2 A_4^T \end{vmatrix}$$

Since the KLT produces N fully decorrelated coefficients, $\sum_z$ is a diagonal matrix. Either both A_2 and A_3, or both A_1 and A_4 must be identically zero. For $A_2 = A_3 = 0$, the first N/2 vectors remain even, while for $A_1 = A_4 = 0$, the even and odd vectors are interchanged in K.

$$K = AH$$

$$= \begin{vmatrix} A_1 & 0 \\ 0 & A_4 \end{vmatrix} \begin{vmatrix} E & \tilde{E} \\ D & -\tilde{D} \end{vmatrix}$$

$$= \begin{vmatrix} A_1 E & A_1 \tilde{E} \\ A_4 D & -A_4 \tilde{D} \end{vmatrix}$$

$$= \begin{vmatrix} A_1 E & A_1 E\tilde{I} \\ A_4 D & -A_4 D\tilde{I} \end{vmatrix}$$

The KLT is an even/odd transform for the class of correlation matrices for which even/odd transforms decorrelate the even and odd vector coefficients.

If H and J are two even/odd NXN transforms, the multiplication matrix for conversion between them is sparse:

$$H = \begin{vmatrix} E & \tilde{E} \\ D & -\tilde{D} \end{vmatrix}$$

$$J = \begin{vmatrix} F & \tilde{F} \\ G & -\tilde{G} \end{vmatrix}$$

The conversion is defined by

$$H = SJ$$

$$S = HJ^T$$

$$= \begin{vmatrix} E & \tilde{E} \\ D & -\tilde{D} \end{vmatrix} \begin{vmatrix} F^T & G^T \\ \tilde{F}^T & -\tilde{G}^T \end{vmatrix}$$

$$= \begin{vmatrix} EF^T + \tilde{E}\tilde{F}^T & EG^T - \tilde{E}\tilde{G}^T \\ DF^T - \tilde{D}\tilde{F}^T & DG^T + \tilde{D}\tilde{G}^T \end{vmatrix}$$

However,

$$\tilde{E}\tilde{G}^T = E\tilde{I}(G\tilde{I})^T = E\tilde{I}\tilde{I}G^T = EG^T$$

$$\tilde{D}\tilde{F}^T = DF^T$$

It follows that

$$S = 2\begin{vmatrix} EF^T & 0 \\ 0 & DG^T \end{vmatrix}$$

The conversion between any two even/odd transforms requires $N^2/2$ rather than N^2 multiplications.

We have shown that the class of even/odd transforms has no correlation between the even and odd vector coefficients, for a class of data correlation matrices including the stationary data matrix. The KLT for this data correlation class, and many familiar transforms, are even/odd transforms. The coefficients of any even/odd transform can be obtained by a sparse matrix multiplication of the coefficients of any other even/odd transform. This observation was the basis of a previous implementation of the DCT and suggested the investigation of even/odd transforms described below.

THE DISCRETE COSINE TRANSFORM OBTAINED VIA THE HADAMARD TRANSFORM

Hein and Ahmed have shown how the DCT vectors can be obtained by a sparse matrix multiplication on the WHT vectors (3, 4). Since the DCT, unlike the general KLT, has a constant vector and a shifted square-wave vector in common with the WHT, the number of matrix multiplications is less than $N^2/2$. The A matrix, which generates the DCT vectors for $N = 8$ from the WHT vectors, is given by Hein and Ahmed, and is reproduced here as Figure 1. Although this implementation of the DCT requires more operations for large N than the most efficient DCT implementation (5), it is very satisfactory for N equal to 8.

If a transform has even and odd vectors and has a constant vector, as is typical, it can be obtained via the WHT in the same way as the DCT. The slant transform is an example (1, 6). A hardware implementation of the DCT via the WHT is being constructed at Ames Research Center, using $N = 8$ and the matrix of Figure 1. Since this implementation contains the matrix multiplication factors in inexpensive read-only memories, it will be possible to consider the real-time quantization design and evaluation of a large class of transforms. Transforms with suboptimum performance are acceptable only if they can be implemented with reduced complexity. Transform performance can be determined theoretically from the vector energy compaction, while the implementation complexity can be estimated from the number and type of operations added after the WHT.

COMPARISON OF TRANSFORMS USING THE FIRST-ORDER MARKOV CORRELATION MODEL

It is generally accepted that the sample-to-sample correlation of an image line scan is approximated by the first-order Markov model (7).

$$c(x_i, x_j) = c(|i - j|) = r^{|i-j|}$$

The correlation of adjacent samples, r, varies from 0.99 for low detail images to 0.80 for high detail images, with an average of about 0.95 (8). The correlation matrix, $\sum_R$, was generated using the first-order Markov model, for various r, and the corresponding KLT's and

vector energies were numerically computed. (The analytic solution is known (9).) In addition, the matrix Σ_x was used to compute the transform vector energies and correlations for the WHT, DCT, and other transforms.

As is well known, the KLT vectors for $r = 0.95$ are very similar to the DCT vectors and have nearly identical vector energies (1, 3). The most apparent difference between the DCT and the KLT is that the KLT vector corresponding to the constant DCT vector is not exactly constant, but weights the central samples in a fixed transform block more than samples near the edge of the block. As r approaches 1.00, this KLT vector approaches the constant vector, and all the KLT vectors approach the corresponding DCT vectors. The vector energies of the KLT and the DCT are nearly identical for r greater than 0.90, and differ only slightly for r greater than 0.50. The KLT and DCT vector energies for $N = 8$ and $r = 0.50$ are plotted in Figure 2. The energy compaction at $r = 0.5$ is much less than at the typical $r = 0.95$.

The rate-distortion performance of a transform depends on the transform energy compaction. If the distortion d is less than the coefficient variance σ_i^2 for all i, all N transform vectors are quantized and transmitted. The number of bits required is (10):

$$b = \sum_{i=1}^{N} \frac{1}{2} \log_2(\sigma_i^2/d)$$

$$= \frac{1}{2} \sum_{i=1}^{N} \log_2 \sigma_i^2 - \frac{N}{2} \log_2 d$$

The first term of b can be used as a figure of merit for a transform.

$$f = \frac{1}{2} \sum_{i=1}^{N} \log_2 \sigma_i^2$$

The figure of merit f is a negative number; the larger its magnitude, the greater the rate reduction achieved by the transform. Table I gives f for the KLT, DCT, WHT, and two even/odd transforms that will be described below. At correlation $r = 0.95$, the KLT gains 0.014 bits more than the DCT and 1.183 bits more than the WHT. The WHT achieves most of the available data compression, and the DCT achieves nearly all. As this rate reduction is obtained for all input vectors, the increased compression of the DCT over the WHT, for $r = 0.95$, is 1.169 - 1.018, or 0.15 bits per sample.

EVEN/ODD TRANSFORMS OBTAINED VIA THE WALSH-HADAMARD TRANSFORM

The sequency of a transform vector is defined as the number of sign changes in the vector. The vector sequencies of the vectors corresponding to the matrix of Figure 1 are in bit-reverse order, as indicated (0, 4, 2, 6, 1, 5, 3, 7). The energy compaction of the WHT and DCT for $r = 0.95$ and $N = 8$ is shown in Figure 3. In the conversion from WHT to DCT, the two-by-two matrix operation on vectors 2 and 6 transfers energy from 6 to 2. The four-by-four matrix operation on the vectors of sequency 1, 5, 3, and 7 reduces the energy of 3, 5, and 7 and increases the energy of 1. These operations remove most of the residual correlation of the WHT vectors. The matrix multiplication requires 20 multiplications by 10 different factors (15 factors including sign differences).

We first consider a simplified operation on the 2 and 6 and the 1 and 3 sequency vectors. This operation consists of multiplying the WHT vectors by matrix B (Figure 4). This further transform is designed to reduce correlation and to generate new transform vectors in a way somewhat similar to the A matrix multiplication which produces the DCT. There are two identical two-by-two operations, and a total of eight multiplications by two different factors (three including sign). The energy compaction of the B-matrix transform is shown in Figure 3, with the energies of the WHT and DCT. As the B-matrix transform vectors of sequency 0, 4, 5, and 7 are identical to the WHT vectors, they have identical energy. The B-matrix transform vectors of sequencies 0, 1, 2, 3, 4, and 6 are identical to the corresponding DCT vectors (0, 4) or very similar. For example, the B-matrix vector of sequency 1 is a slanted vector of step width 2 and step size 2 (3, 3, 1, 1, -1, -1, -3, -3). The performance of the B-matrix transform, in terms of the figure of merit, is given in Table I above. The B-matrix transform has something more than one-half of the gain of the DCT over the WHT, with something less than one-half of the multiplications, and less than one-fourth the hardware if the two-by-two transformer is used twice.

As a second example, suppose that it is desired to approximate the DCT by adding integer products of the WHT vectors. For small integers, this operation can be implemented by digital

shifts-and-adds, and requires fewer significant bits to be retained. The matrix C, given in Figure 5, is an orthonormal transform matrix that is similar to the DCT. The two-by-two matrix, operating on the vectors of sequency 2 and 6, is a specialization of the general two-by-two matrix having orthogonal rows with identical factors. The four-by-four operation on the vectors of odd sequency is a specialization of the general four-by-four matrix with orthogonal rows, identical factors, and the additional requirement of a positive diagonal.

The specializations of the general matrices were made by requiring that the two-by-two matrix integers have approximately the ratios found in the second (and third) rows of the A matrix, and that the four-by-four matrix integers have approximately the ratios found in the fifth (and eighth) rows of the A matrix. Since the A-matrix transform is the DCT, it is ensured that the C transform vectors of sequency 2, 6, 1, and 7 will approximate the corresponding DCT vectors.

The energy compaction results of the C transform, with the results of the WHT and DCT, are given in Figure 6, for $r = 0.95$ and $N = 8$. The energy of the vectors of sequency 2, 6, 1, and 7 is very similar to the energy of the DCT vectors, but the vectors of sequency 3 and 5 are different. The energy correspondence could be improved by matching the four-by-four matrix factors to the average of the fifth and sixth rows in the A matrix, but there is little potential data compression remaining. The theoretical performance of the C matrix, in terms of the figure of merit, is given in Table I. The C-matrix transform obtains nearly all the gain of the DCT over the WHT. If the rational form, instead of the integer form, of the C-matrix transform were used, the computation would require 16 multiplications by 4 different factors (7 factors including sign differences). There is some reduction in complexity from the implementation of matrix A.

EXPERIMENTAL IMAGE COMPRESSION RESULTS

Experimental results were obtained for two-dimensional, 8x8 sample block implementations of the transforms considered above. Four video test images — Harry Reasoner, two Girls, two Men, and band — were used in all tests. These images have correlation of 0.97 to 0.98 between elements in the scan line, and fit the first order Markov model, except for the very detailed band image, which deviates from the Markov model and has an average in-line correlation of 0.85 (11). Two different compression experiments were made.

The test images were first compressed by representing either thirty-two or sixteen of the sixty-four 8x8 transform vectors, using an eight-bit uniform, full-range quantizer. The other vectors were neglected. The patterns of the vectors transmitted and neglected are given in Figure 7. The vectors are in sequency order, with the lowest sequency average vector in the upper left corner of the pattern. The mean-square error for this compression method and the four transforms is given in Table II. The B-matrix transform error is intermediate between the WHT and DCT errors, and the C-matrix error is very close to the DCT error. This is consistent with the Markov model energy compaction results above.

To obtain the greatest transform compression, the transmitted bits should be assigned to the vectors according to the equation given above, and the coefficient quantizers should be designed for minimum error given the coefficient energy and amplitude distributions. The optimum theoretical bit assignments and quantizers depend on the particular transform used. The test images, and most typical images, contain low-contrast, high-correlation background areas, and edges where correlation is low. The bit assignments and quantizer designs based on the stationary Markov model ignore this nonstationarity, and designs that consider low-contrast areas and edges give improved mean-square error and subjective performance. Such improved designs have been devised for the WHT (11), and have been tested with the DCT, B-matrix, and C-matrix transforms. The transmission rate and mean-square error results are given in Figure 8, for the test images compressed in the video field. The DCT gives improved error performance, and the B and C matrix transforms are intermediate, but the B and C matrix results are relatively poorer than those in Table II. The DCT gives more rate reduction than the WHT — about 0.2 to 0.5 bits per sample. As a two-dimensional transform has twice the gain of a one-dimensional transform (.0), the theoretical gain of the DCT over the WHT, for $r = 0.95$, should be twice the 0.15 bits per sample of Table I, or 0.30 bits per sample.

The lower error of the DCT, B-matrix, and C-matrix transforms does indicate subjective improvement in the compressed images. This subjective improvement is larger at lower total bit rates, due to the relative increase of larger, more noticeable errors at the lower rates, and due to the more objectionable, blocky nature of large WHT errors. The B and C matrix errors are subjectively more similar to the DCT errors than to WHT errors, because the higher energy vectors approximate the DCT vectors.

It is not surprising that a design optimized for the WHT gives good results for the DCT and similar transforms. The transform compression introduces errors in three ways: not transmitting vector coefficients, by using quantizers that are too narrow, and quantization

errors within the quantizer ranges. The DCT, because of its superior energy compaction, reduces the first two sources of error. Although the quantizers used are nearly uniform, they do have smaller quantization steps for low coefficient values, so the third source of error is also reduced. Any compression design will give better performance with the DCT. From the similarity in energy compaction, a good design for the WHT should be reasonably effective for the DCT. However, further performance gains can be made with the DCT and with the R-matrix and C-matrix transforms, by optimizing the compression designs for the transforms used.

The error statistics show that the lower mean-square error of the DCT is due both to fewer large errors, which nearly always occur at edges, and to fewer small errors, which occur in flat areas and edges. The subjective appearance of the compressed image confirms that the DCT produces both smoother low contrast areas and less distorted edges. Since the low contrast areas have very high correlation, and since the edges — though not noise-like — can be approximated by a low-correlation Markov model, the mean-square error and subjective results agree with the theoretical result that the DCT is superior to the WHT for high values of correlation (see Table I).

CONCLUSION

The Karhunen-Loeve transform for data with stationary correlation, the discrete cosine transform, the Walsh-Hadamard transform, and other familiar transforms are even/odd vector transforms whose coefficients can be obtained by sparse matrix multiplications of the coefficients of other even/odd transforms. Of the familiar transforms, the Walsh-Hadamard transform is the simplest to implement, but has the smallest compression gain. Using the Walsh-Hadamard transform followed by a sparse matrix multiplication allows implementation of any even/odd transform. The discrete cosine transform has a difficult implementation, but very closely approaches the optimum performance for first-order Markov data. As the form of the vectors is modified to approach that of the discrete cosine vectors, the vector energy compaction and the theoretical and experimental image compression results approach those of the discrete cosine transform. The theoretical data compression reliably indicates the difference in experimental performance for these transforms. About one-half of the performance difference between the Walsh-Hadamard and the discrete cosine transforms can be achieved by simple post processing of the Walsh-Hadamard coefficients.

ACKNOWLEDGMENTS

The authors are grateful to D. R. Lumb and L. B. Hofman of Ames Research Center, and to N. Ahmed of Kansas State University, for their assistance and encouragement. The authors are also grateful for the use of the Ames Research Video Compression Facility.

REFERENCES

1. Ahmed, N. and Rao, K. R., Orthogonal Transforms for Digital Signal Processing. Springer-Verlag, Berlin, 1975.
2. Berger, T., Rate Distortion Theory. Prentice Hall, Englewood Cliffs, New Jersey, 1971.
3. Ahmed, N., Natarajan, T., and Rao, K. R., "Discrete Cosine Transform," IEEE Trans. Computers. Vol. C-23, Jan. 1974, pp. 90-93.
4. Hein, D. and Ahmed, N. "On a Real-Time Walsh-Hadamard/Cosine Transform Image Processor," IEEE Trans. Electromagnetic Compatibility. Vol. EMC-20, Aug. 1978, pp. 453-457.
5. Chen, W. H., Smith, C. H., and Fralick, S. C., "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Trans. Communications. Vol. COM-25, Sept. 1977. pp. 1004-1009.
6. Pratt, W. K., Chen, W. H., and Welch, L. R., "Slant Transform Image Coding," IEEE Trans. Communications. Vol. COM-22, Aug. 1974, pp. 1075-1093.
7. Franks, L. E., "A Model for the Random Video Process," Bell Syst. Tech. J., April 1966, pp. 609-630.
8. Connor, D. J. and Limb, J. O., "Properties of Frame-Difference Signals Generated by Moving Images," IEEE Trans. Communications. Vol. COM-22, Oct. 1974, pp. 1564-1575.
9. Jain, A. K., "A Fast Karhunen-Loeve Transform for a Class of Random Processes," IEEE Trans. Communications. Vol. COM-24, Sept. 1976, pp. 1023-1029.
10. Davisson, L. D., "Rate-Distortion Theory and Application," Proc. IEEE. Vol. 60, July 1972, pp. 800-808.
11. Jones, H. W., "A Comparison of Theoretical and Experimental Video Compression Designs," to appear in IEEE Trans. Electromagnetic Compatibility, Feb. 1979.

TABLE I The Figure of Merit, $t = \sum \log_{10} \sigma_k^2$, for Different Transforms at $N = 8$ and Various Correlations.

	Transform				
Correlation, r	KLT	DCT	WHT	B matrix	C matrix
0.99	-19.817	-19.775	-18.499	-19.205	-19.597
0.95	-11.743	-11.729	-10.560	-11.206	-11.558
0.90	-8.179	-8.141	-7.111	-7.875	-8.180
0.80	-5.162	-5.092	-4.317	-4.731	-4.954
0.70	-3.402	-3.328	-2.765	-3.056	-3.214
0.50	-1.453	-1.396	-1.136	-1.261	-1.333
0.00	0.00				

TABLE II The Mean-Square Error for the WHT, DCT, B Matrix and C Matrix Transforms with a Subset of Vectors Retained.

Mean-square error for 32 vectors retained

	Reasoner	Two Girls	Two Men	Band
WHT	0.558	0.806	1.694	3.948
B matrix	0.500	0.738	1.581	3.628
C matrix	0.442	0.666	1.536	3.310
DCT	0.446	0.660	1.535	3.056

Mean-square error for 16 vectors retained

	Reasoner	Two Girls	Two Men	Band
WHT	1.619	2.206	4.801	12.322
B matrix	1.507	2.093	4.557	12.056
C matrix	1.427	2.029	4.447	11.897
DCT	1.430	2.031	4.406	11.628

$$A = \begin{bmatrix} 1.0 & & & & & & & \\ & 1.0 & & & & & & \\ & & 0.923 & 0.383 & & & & \\ & & 0.383 & 0.923 & & & & \\ & & & & 0.907 & 0.075 & 0.375 & 0.180 \\ & & & & 0.214 & 0.768 & 0.513 & 0.318 \\ & & & & 0.318 & 0.513 & 0.768 & 0.214 \\ & & & & 0.180 & 0.375 & 0.075 & 0.907 \end{bmatrix}$$

Figure 1 - The A Matrix Used to Obtain the DCT From the WHT.

C - 34

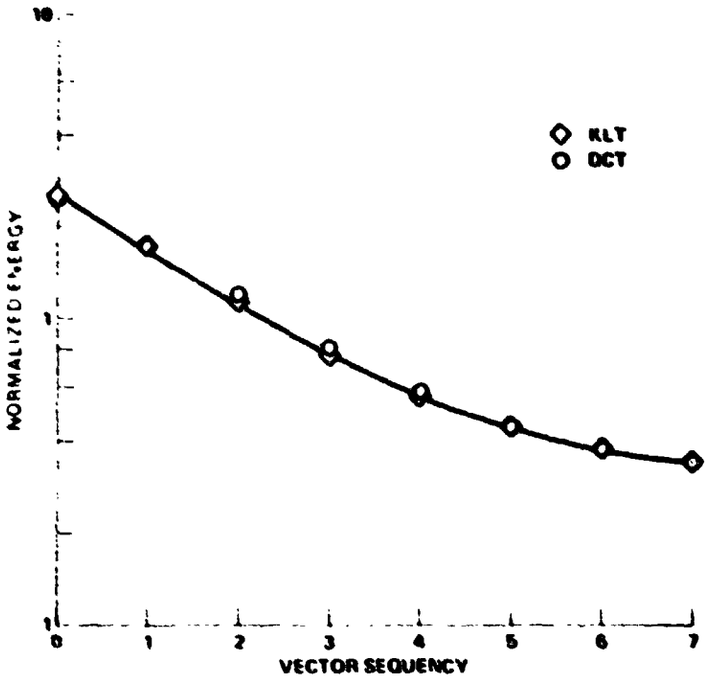
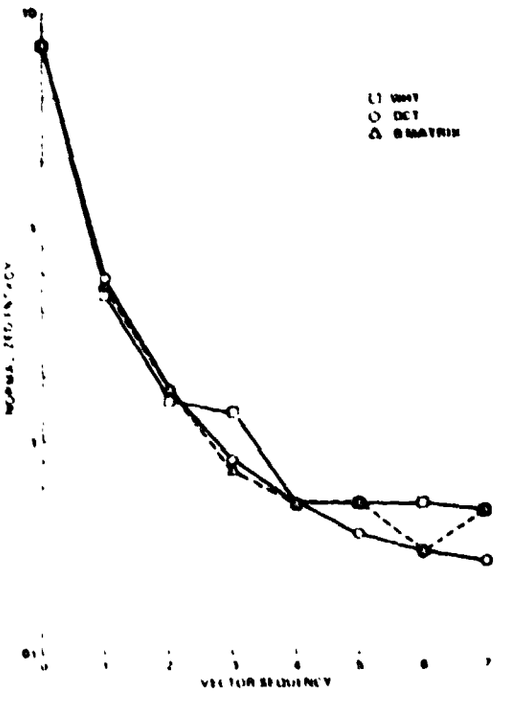


Figure 2 KLT and DCT Vector Energies for N = 8 and r = 0.9.



$$B = \begin{bmatrix} 1.00 & & & & & & & \\ & 1.00 & & & & & & \\ & & 2/\sqrt{5} & 1/\sqrt{5} & & & & \\ & & 1/\sqrt{5} & 2/\sqrt{5} & & & & \\ & & & & 2/\sqrt{5} & 0 & 1/\sqrt{5} & 0 \\ & & & & 0 & 1.00 & 0 & 0 \\ & & & & 1/\sqrt{5} & 0 & 2/\sqrt{5} & 0 \\ & & & & 0 & 0 & 0 & 1.00 \end{bmatrix}$$

Figure 4 The B Matrix.

$$C = \frac{1}{13}\begin{bmatrix} 13 & & & & & & & \\ & 13 & & & & & & \\ & & 12 & 5 & & & & \\ & & 5 & 12 & & & & \\ & & & & 12 & 0 & 4 & 3 \\ & & & & 0 & 12 & 3 & 4 \\ & & & & 4 & 3 & 12 & 0 \\ & & & & 3 & 4 & 0 & 12 \end{bmatrix}$$

Figure 5 The C Matrix.

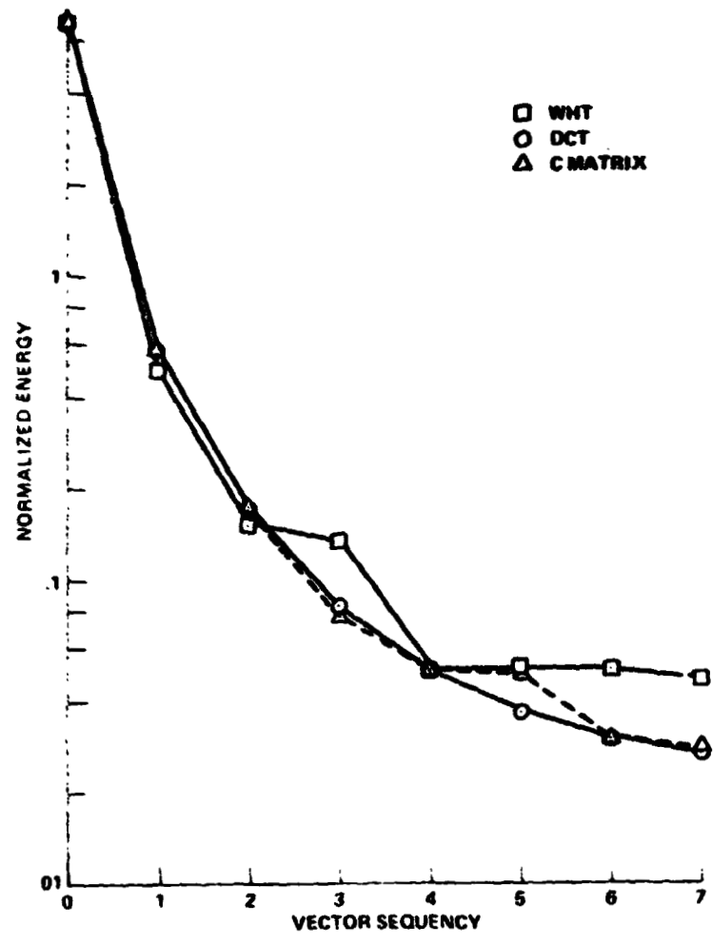


Figure 6 - The Energy Compaction of the WHT, DCT, and C Matrix Transforms for N = 8 and r = 0.95.

8	8	8	8	8	8	8	8
8	8	8	8	8	8	8	8
8	8	8	8	0	0	0	0
8	8	8	8	0	0	0	0
8	8	0	0	0	0	0	0
8	8	0	0	0	0	0	0
8	8	0	0	0	0	0	0
8	8	0	0	0	0	0	0

32 vector pattern

8	8	8	8	8	8	8	8
8	8	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0

16 vector pattern

Figure 7 - Patterns of Vector Coefficients Retained and Neglected

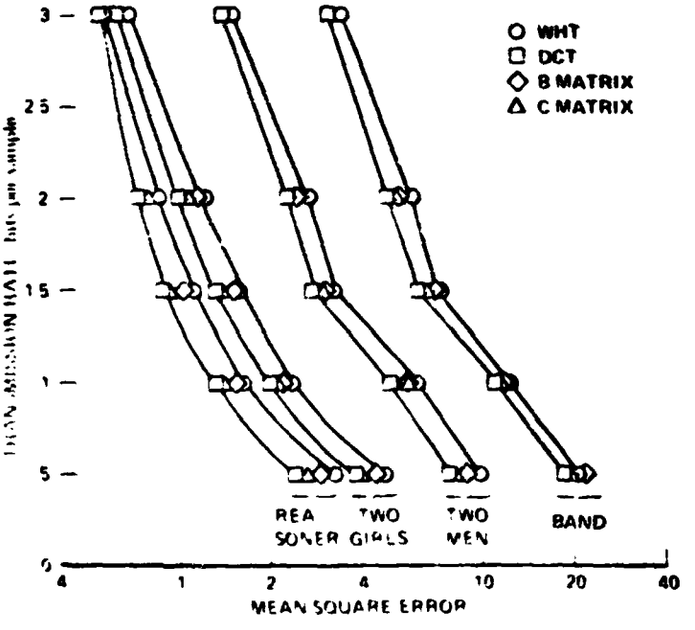


Figure 8 Transmission Rate Versus Error for the Four Test Images