Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data

ROBERT F. RICE AND JAMES R. PLAUNT

***Abstract*—An adaptive variable length coding system is presented. Although developed primarily for the proposed Grand Tour missions, many features of this system clearly indicate a much wider applicability.**

Using sample to sample prediction, the coding system produces output rates within 0.25 bit/picture element (pixel) of the one-dimensional difference entropy for entropy values ranging from 0 to 8 bit/pixel. This is accomplished without the necessity of storing any code words. Performance improvements of 0.5 bit/pixel can be simply achieved by utilizing previous line correlation.

A Basic Compressor, using concatenated codes, adapts to rapid changes in source statistics by automatically selecting one of three codes to use for each block of 21 pixels. The system adapts to less frequent, but more dramatic, changes in source statistics by adjusting the mode in which the Basic Compressor operates on a line-to-line basis. Furthermore, the compression system is independent of the quantization requirements of the pulse-code modulation system.

I. Introduction

A GRAND TOUR of the outer planets (Jupiter, Saturn, etc.) will be undertaken in the late 1970's and early 1980's [1]. Image data on these missions will be of major scientific importance since television pictures taken up to 200 days from planetary encounter will provide valuable new information. Conceptually we can expect data compression to increase the number of scientifically valuable images returned from the spacecraft. The primary content of this paper is concerned with the development of a television data compression system [2]–[6] compatible with the unique requirements and constraints of the Grand Tour missions. In addition, some generalizations of this basic system will be noted.

Image data is used both for photointerpretive purposes and quantitative measurements. Quantitative uses include the measurement of geometric relationships between identifiable features and the precise measurement of scene luminances or colors such as for atmospheric scattering studies. The quantitative measurements place the more severe constraints on the camera system design and essentially led to the basic assumption of linear quantization of the camera analog output. In particular, the baseline data system presently considered at the Jet Propulsion Laboratory will output conventional pulse-code modulation (PCM) data with 8-bits/picture element (pixel), 800 pixels/line and 800 lines/picture.

Manuscript received May 24, 1971; revised July 16, 1971. This work was supported by NASA under Contract NAS7-100.
The authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Calif. 91103.

Clearly, the difficulties in reliably transmitting the high rate television data from such great distances are severe. Real-time transmission capability is severly constrained at the more distant planets: Uranus, Neptune, and Pluto. Under these conditions it becomes necessary to include an on-board mass memory to store data for later transmission. This mode of operation is particularly important during the time-constrained period of nearest planetary encounter. For reliability reasons, this on-board storage medium should be sized to hold a minimum acceptable number of PCM television pictures. The primary advantage of data compression here is to increase the number of stored pictures beyond this minimum. In the case of real time transmission the advantage of data compression is to increase the number of pictures which can be transmitted in a fixed time interval.

It is well known that redundancy removal techniques, such as that to be developed here, tend to propagate errors. Hence it becomes necessary to transmit compressed data at a lower error rate than uncompressed data. This results in an error rate (quality) versus transmission rate tradeoff for real-time transmission. However, the significance of this tradeoff is greatly reduced in the case of stored compressed data since this data can be transmitted back at a leisurely transmission rate which obtains the desired error rate. Although these tradeoffs are important, noiseless channel conditions (approximating the stored data case) will henceforth be assumed unless noted otherwise.

Initial Constraints

An imaging experiment is by its very nature exploratory and particularly for distant planets, the visual properties of the object scene are not known in any detail. This fact, along with the considerations leading to the linear quantization assumption, imply that the approach to television data compression should be conservative so as not to destroy or distort significant, albeit unexpected, picture detail. Thus the first developmental constraint placed on a Grand Tour compressor was that it be information preserving in the sense that the camera PCM data can be exactly reconstructed.

Because of anticipated memory limitations, a second constraint imposed on the data compressor was that it be restricted to one-dimensional processing.

Of course, implicit with the design of any spacecraft subsystem is the requirement that the implementation be compatible with the "power, weight, volume" limitations of the spacecraft.

Sections II–IV are devoted to the development of a television data compression system subject to the initial constraints described above. The reader should keep in mind the applicability of these results to the more general source coding problem. A first step in extending these results is taken in Section V by utilizing line-to-line correlation to improve performance.

II. Performance Measures

The initial constraints for Grand Tour imaging essentially dictated an approach to compression which utilizes the concept of variable length coding of PCM data on a line by line basis. The functional operation and performance of the resulting coder are treated in later sections. This section is devoted to the description of an appropriate data model and performance criteria.

Data Model

The Grand Tour compressor must operate over an extraordinarily wide range of anticipated images. Any data model must of necessity take this into account. The approach here will be to make certain practical simplifying assumptions on the local details of an image in order to more realistically model the global nonstationarity of the image data source as a whole. More specifically, the model will treat differences between adjacent pixels as a discrete memoryless source but will permit the distribution of differences to change throughout each picture.

Fig. 1 illustrates one possible configuration for a variable length coding system. The random variable x_j denotes the jth sample in a data stream, taking on the quantization values $0, +1, +2, \cdots, 2^n - 1$. $\hat{x}_j$ denotes a prediction of x_j based on all preceeding data. As shown in the diagram, a variable length coder is applied to the difference[1], $d_j = x_j - \hat{x}_j$.

The first practical limitation on this system is that $\hat{x}_j$ can be based on, at most, those data samples which have preceeded x_j along the same scan line. Furthermore, correlation studies on Surveyor and Mariner pictures indicated that, under this limitation, only x_{j-1} contributed significantly to the prediction process. In fact, except for rather atypical situations, the best prediction of x_j is simply x_{j-1}. Thus the coding problem reduces to coding the differences between adjacent samples, and modeling the data reduces to characterizing these differences.

Difference Distributions: Let d be the difference between two adjacent pixels. In an n-bit PCM system d can take on the values $0, \pm 1, \pm 2, \cdots, \pm(2^n - 1)$. Adjacent samples of d are assumed to be statistically independent, and the distributions of d, denoted by P, are assumed to have the following property:

$$\Pr[d = 0] \geq \Pr[d = +1] \geq \Pr[d = -1] \geq \Pr[d = +2] \cdots \geq \Pr[d = -(2^n - 1)]. \quad (1)$$

The consistency of (1) as a good approximation to actual observed distributions is an important reason for using differences as part of a coding system. Half of the job in developing an efficient variable length code for any distribution is to assign the shortest code words to source symbols which have the greatest probability of occurrence and the longest to those symbols which have the smallest probability of occurrence. The best *a priori* ordering of code words is thus supplied by condition (1).

To take into account the nonstationary nature of television images we first divide each sampled television line into equal length blocks of J successive differences (J will be specified shortly). For identification purposes, these blocks can be labeled consecutively from left to right across the first line, followed by the second line, and so on. We assume that a possibly different distribution drives each block. Thus let $P[k]$ denote the difference distribution for the kth block, where k runs from 1 to N, the number of blocks in the area or picture under consideration. The composite distribution for an area is given by

$$P^* = \frac{1}{N} \sum_k P[k]. \quad (2)$$

A *uniformly active* area is defined as one for which the same probability distribution holds throughout. With $P[k] = P$ for all k, we see that $P^* = P$.

Performance Lower Bounds [7], [8]: The entropy for distribution P is given by

$$H(P) = -\sum_i \Pr[d = i] \log \Pr[d = i] \quad (3)$$

where the log function is assumed to be of base 2. By Shannon's theorem, if a discrete memoryless source genrates symbols according to distribution P, then no coding scheme which results in unambiguous decoding can result in an average coding rate which is less than $H(P)$ bits/sample.

It is felt that the zero memory assumption for differences is a good one so that $H(P)$ can be used as a good lower bound to the performance of a difference coder when the difference distribution is P.

Let $L(P)$ denote the number of bits/pixel required by a particular coder to code a sequence of difference samples which has resulted from distribution P. The conclusion of the preceding paragraph may be stated mathematically as

$$E[L(P)] \geq H(P) \quad (4)$$

where E denotes expectation. Note that this holds for a sequence of any length, provided that each difference sample has resulted from distribution P. For this special case, a coder can be said to "perform well, relative to the data" if the difference between $E[L(P)]$ and $H(P)$ is

[1] This is already a simplification from the more general system in which an estimator would estimate the distribution of x_j, $\Pr[x_j | x_{j-1}, x_{j-2}, \cdots]$, and the coder would select and use from its memory the variable length code most compatible with $\Pr[x_j | x_{j-1}, x_{j-2}, \cdots]$.

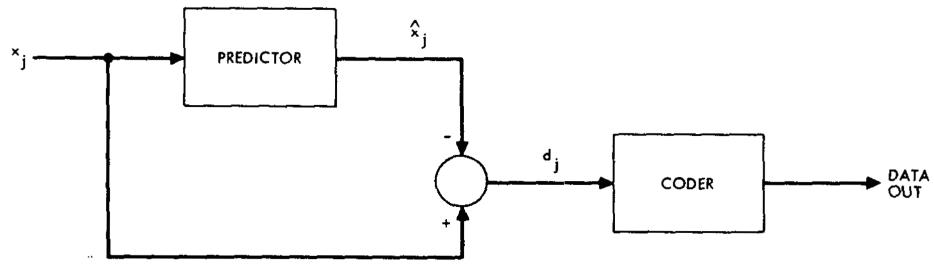

Fig. 1. Coding system.

small. We will be especially interested in sequences of length J which make up a block.

The Huffman coding algorithm will produce a code which will perform well in this sense. However, such a code is optimized only for the one distribution P (it may also be the best code for other distributions which are similar to P). We have already strongly implied in preliminary discussions that a rather broad spectrum of difference distributions can be expected, or at least must be assumed. It is easy to cite examples where the average performance of a single code is close to optimum for one P but several bits above the entropy for another P. Thus, a single code is clearly inadequate.

The generality of our model assumes that P can change completely from block to block. This makes the analysis tractable, since each block of data can be thought of independently of all others. As already implied, it is questionable whether a more detailed analysis involving higher order stochastic processes is either possible or useful considering the extremely wide variety of data considered here (in a later section we will consider $H(P)$ values out to 8 bits/pixel).

The average performance of a particular coding algorithm on N independent blocks is defined as

$$\hat{L} = \frac{1}{N} \sum_{k=1}^{N} L(P[k]). \tag{5}$$

The N blocks could make up an area or a complete picture.

Using the definition of entropy on each block of data we define the average entropy for N independent blocks as

$$\hat{H} = \frac{1}{N} \sum_{k=1}^{N} H(P[k]). \tag{6}$$

Note that $\hat{H}$ is a single number, whereas $\hat{L}$, like $L(P[k])$, is a random variable. The distribution $P[k]$ which drives the kth block can result in many different sequences. $L(P[k])$ takes on values equal to the number of bits/pixel required to code these resulting sequences.

Applying (4) to each block, we find that $\hat{H}$ is a lower bound to the expected performance of any coding algorithm for a complete area or picture. This can be stated mathematically as

$$\Delta = E[\hat{L}] - \hat{H} \geq 0 \tag{7}$$

or equivalently

$$\frac{1}{N} \sum_{k=1}^{N} \{E[L(P[k])] - H(P[k])\} \geq 0. \tag{8}$$

Clearly, a coding algorithm performs efficiently if Δ is small. Inequality (8) points out how this might be achieved since Δ must be small if every term in (8) is small. This would appear to present an insurmountable verification problem, since to know how well one has performed in a certain block implies a knowledge of the distribution driving that block. In general this knowledge is not available since a sequence of, say, 21 samples hardly constitutes enough data to validly determine a distribution. However, if we can show that $E[L(P)] - H(P)$ is small for every typical distribution[2] which can occur, then it becomes immaterial just which distribution $P[k]$ is. In this case, each term in (8) is always small so that the particular algorithm is always efficient. The next section will deal with showing that this result holds for the Basic Compressor for the Principal Operating Range of entropy from 0 to 4 bits/pixel.

Another useful term, and one which is readily calculable, is the entropy of the composite distribution P^*. $H(P^*)$ can be interpreted as a lower bound to the expected performance of a nonadaptive coding scheme. That is, one which does not take into account the possible changes in P which may be present in a picture. Due to the convexity of the entropy function

$$\hat{H} \leq H(P^*). \tag{9}$$

It is easy to choose examples where the difference between $\hat{H}$ and $H(P^*)$ is considerable (over 1 bit/pixel). This clearly points out the advantage of an adaptive coding scheme. It should also alleviate any anxiety when we later produce some performance results where $E[\hat{L}] < H(P^*)$.

III. Basic Compressor

Functional Operation [4]

A block diagram of the Basic Compressor is shown in Fig. 2. Each block of J pixels enters a Difference Converter which computes differences between adjacent

[2] By "typical distribution" we mean one which is truly representative of Grand Tour television data and at least approximates (1).

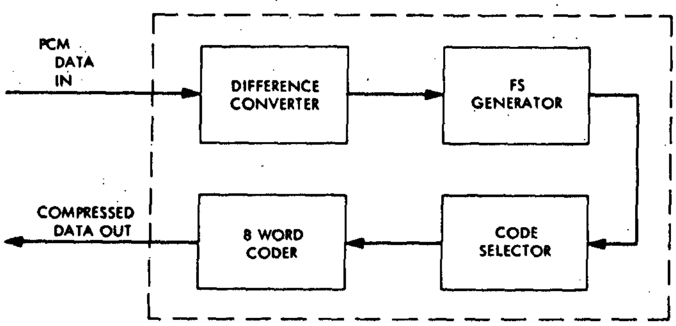

Fig. 2. Basic Compressor.

pixels. The next step is the generation of a Fundamental Sequence (FS) with a simple variable length code. The FS enters a Code Selector which decides to either 1) transmit the FS unmodified, or 2) code the FS with a second variable length code consisting of 8 code words, or 3) first complement each bit of the FS, then code it with the same 8-word code. These operations are performed separately on each data block.

Difference Converter: The Difference Converter of the Basic Compressor can be implemented as a simple arithmetic unit, producing blocks of J differences as signed binary words. Each block then enters the FS Generator for further coding.

FS Generator[3]: The FS Generator is a variable length coder particularly chosen for the simplicity. The ith code word is defined as

$$u_i \equiv \overbrace{000 \cdots 000}^{(i-1)\text{ zeroes}}1. \tag{10}$$

Each of the J difference samples are individually coded with this code according to the assignment in Table I. The sequence of bits formed by concatenating the J resulting code words forms an FS.

The coding described by Table I can be easily implemented with a binary counter and comparator. Note that the form of this implementation is essentially independent of the possible number of differences and hence independent of the quantization requirements of the input PCM data. This will become an important point in a later section.

It is of some interest to note that an FS will always have exactly J ones and will always end in a one.

Code Selector: As shown in Fig. 2, an FS enters a Code Selector which decides to either 1) transmit the FS unmodified, or 2) code the FS further with an 8-word binary code (the resulting sequence is called Code FS) or 3) first complement each bit of the FS, then code it with the same code as in 2) (called Code $\overline{\text{FS}}$). The coding operations in 2) and 3) will be discussed subsequently.

The decision to use options 1), 2), or 3) is based on

[3] Another form of FS is discussed in [2] and [3].

TABLE I
FS Generator Code

Difference d	Code Word u_i
0	1
+1	01
−1	001
+2	0001
−2	00001
·	·
·	·

the length of the FS in bits $\hat{F}$ shown in Table II. The Transmission Criterion can be simply implemented by a shift register with $3J$ stages [4]. The motivation for the selection of the thresholds in Table II is beyond the scope of this paper. A thorough treatment is provided in [5].

8-Word Coder: When the Code Selector decides to use the options Code FS or Code $\overline{\text{FS}}$, the 8-word coder is presented with a sequence of $\hat{F}$ binary digits (either the FS itself or $\overline{\text{FS}}$ respectively). The 8-word Coder codes consecutive 3-tuples making up this sequence according to the assignments in Table III. If three does not divide $\hat{F}$ then dummy bits are added.

The 3 tuples can be interpreted as addresses to a memory containing the corresponding 8 code words. Better yet, the small number of code words makes it feasible to implement the 8-word Coder as a sequential machine. In either case, the implementation requirements are rather minimal.

Dynamic Performance

Here we investigate the performance of the Basic Compressor for all typical difference distributions P, such that $0 \leq H(P) \leq 4.5$. The sources of data for this investigation were some 30 Surveyor 7 and Mariner 69 pictures along with several test pictures taken with the Mariner 71 camera system.

Areas of Surveyor pictures (105 by 105) were selected as being uniformly active in the sense that the same kind of data existed throughout. By an earlier definition, the same P drives each block in each such area. The 105 by 105 data samples provide an adequate number of samples to compute $P = P^*$, $E[L(P)]$, and $H(P)$ with some statistical confidence.

These areas were chosen such that their $H(P)$ values covered the range from 0 to 4.5 bits/pixel. The resulting plot of $E[L(P)]$ versus $H(P)$ is shown for the Basic Compressor in Fig. 3 with the optimum block size of $J = 21$.[4]

[4] The $E[L(P)]$ curve includes 2 bits of ID per data block to identify the choice of Code $\overline{\text{FS}}$, FS, or Code FS. For value of J less than 21 the cost of these ID bits increases rapidly, adversely affecting overall performance. Performance is rather insensitive to increases in block size above 21, but clearly implementation costs increase with increasing block size. In this sense $J = 21$ was determined as the optimum block size and will henceforth be assumed.

TABLE II
TRANSMISSION CRITERION

FS Length	$J \le \hat{F} < \frac{3}{2}J$	$\frac{3}{2}J \le \hat{F} < 3J$	$\hat{F} \ge 3J$
Transmission Decision	code $\overline{\mathrm{FS}}$	send FS	code FS

TABLE III
8-WORD CODE

3 Tuple	Code Word
0 0 0	0
0 0 1	1 0 0
0 1 0	1 0 1
1 0 0	1 1 0
0 1 1	1 1 1 0 0
1 1 0	1 1 1 0 1
1 0 1	1 1 1 1 0
1 1 1	1 1 1 1 1

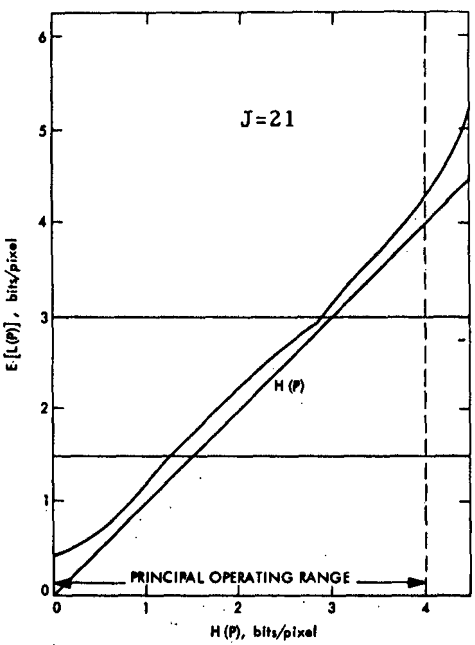

Fig. 3. Basic Compressor performance.

How typical is a typical distribution selected in this manner? That is, given distribution P with entropy $H(P)$, how representative are the numbers $E[L(P)]$ of Fig. 2 for all other distributions which are likely to occur and which also have entropy $H(P)$?

We found that when the quantization requirements were varied from 4 bits/pixel up to 8 bits/pixel and the source data changed from Surveyor 7 to Mariner 69 to Mariner 71 the observed values of $E[L(P)]$ (for a given entropy) fell accurately on the curve of Fig. 3. Although this was not an exhaustive study and special cases exist, it seems fair to conclude the following. Let $\{P\}$ be the set of all typical distributions with entropy $H(P)$. Let P_1 and P_2 be any two distributions in this set. Then $E[L(P_1)] \approx E[L(P_2)]$. Thus Fig. 3 accurately describes the performance of the Basic Compressor for all typical distribution, P, where $0 \le H(P) \le 4.5$ bits/pixel.

The range of entropy values from 0 to 4 bits/pixel is defined as the Principal Operating Range for the Basic Compressor. We note from Fig. 3 that except for a small region near zero entropy, the difference between $E[L(P)]$ and $H(P)$ is on the order of 0.2 bit/pixel or less. Thus, the desired conclusion is that Δ of (7) will be approximately 0.2 bit/pixel if $H(P[k])$ is limited to this range. Improved efficiency for entropy values in excess of 4 bits/pixel is the subject of a later section.

Nonuniform Pictures

To illustrate its adaptive character, the Basic Compressor was applied to the two nonuniform, 6 bit/pixel Surveyor pictures shown in Figs. 4 and 5. $\hat{L}$, the actual performance, and $H(P^*)$, the lower bound for a nonadaptive coding scheme, are shown for both pictures in Table IV.[5] We note that in both cases $\hat{L} < H(P^*)$.

Sensor Noise: The sensor noise of the Surveyor camera system contributed about 1.5 bits/pixel to the entropy of a blank field. Using a Gaussian model [5], estimates of $\hat{L}$ and $H(P^*)$ were made for Surveyor pictures 1 and 2 under the assumptions of zero sensor noise. These figures are also shown in Table IV. We note that $\hat{L}$ is 0.24 bit/pixel less than $H(P^*)$ in both cases. In addition, sensor noise hurts performance by about 0.6 bit/pixel.

IV. IMPROVED PERFORMANCE FOR VERY ACTIVE DATA

By very active data we mean data with entropies greater than 4 bits/pixel, that is, outside the Principal Operating Range of the Basic Compressor. This is a very real possibility for a good portion of the Grand Tour data since quantization of up to 8 bits/pixel is currently planned.

In this section we introduce additional sets of operating modes which effectively extend efficient performance of the Basic Compressor to data with entropy values outside of the Principal Operating Range. These additional modes have other advantages also. The combination of the Basic Compressor with these additional modes has been called *the Rice Machine* in [5]. This terminology will be maintained.

Split-Pixel Definition

The operation of the Basic Compressor is independent of the level of quantization and thus can be made to treat incoming 8-bit data as n-bit data (where $n \le 8$) simply by shifting out the 8-n least significant bits of each data sample. The Split-Pixel option is defined as the capability to compress the n most significant bits of each 8-bit PCM sample with the Basic Compressor (where n = 4, 5, 6, 7 or 8) and to transmit back directly the most

[5] We could also estimate $\hat{H} \approx \hat{L} - 0.2$.

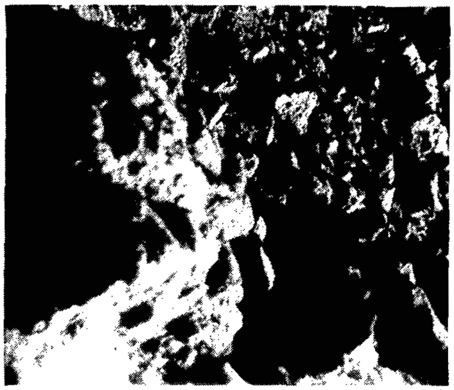

Fig. 4. Surveyor picture 1.

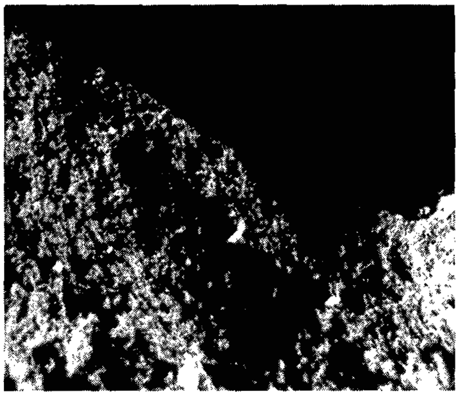

Fig. 5. Surveyor picture 2.

TABLE IV
PERFORMANCE ESTIMATES[a]

	Observed Performance		Estimates Without Noise	
	Basic Compressor Performance	Nonadaptive Lower Bound	Basic Compressor Performance	Nonadaptive Lower Bound
Picture	$\hat{L}$	$H(P^*)$	$\hat{L}$	$H(P^*)$
1	2.90	2.99	2.28	2.52
2	3.00	3.15	2.46	2.70

[a] All figures are in bits/pixel

significant k of the remaining 8-n bits (where $k = 0, 1, \cdots, 8 - n$). These will henceforth be called the K-bits. If k is less than 8-n, then the least significant 8-n-k bits are omitted from transmission altogether. The 15 possible modes are designated by the notation (n, k). Illustrative examples are shown in Fig. 6.

The value of being able to delete bits should be immediately apparent: quantization requirements may change. In a single camera this may be effected before launch as a design change or during the mission(s) if increasing sensor noise makes the least significant bit(s) useless. Also, multiple camera systems on-board the spacecraft (e.g., science and approach guidance) may have differing quantization requirements.

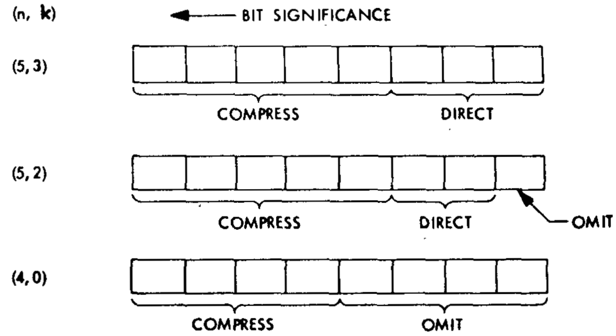

Fig. 6. Split-pixel examples.

Backup-PCM Definition

The Backup-PCM option defined for an (n, k) Split-Pixel mode is defined as follows for a block size of 21:

Transmit directly the most significant $(n + k)$ bits of each pixel in a block whenever the sum of compressed bits plus $21k$ exceeds $21(n + k)$, or equivalently, if the compressed sequence exceeds $21n$ bits.

This option has been included as part of the Rice Machine primarily to prevent data expansion in the presence of atypical data such as fiducial marks (although it can also improve performance in very active data). Backup PCM may be considered as a fourth code along with Code $\overline{\text{FS}}$, FS, and Code FS. The same two ID bits which are used to specify the Basic Compressor's selection of codes are also used to specify when this fourth choice is made.

Formatting Data

This is a good point to describe the data formatting. It should add some clarity to an understanding of the overall compressor operation.

Block Format: Recall that a sequence of data from the Basic Compressor can be one of three forms: Code $\overline{\text{FS}}$, FS, or Code FS. We have added one more possibility at the block level with the inclusion of Backup PCM. Thus, as already stated, the first two bits of each block, called ID, will designate which of the four alternatives have been selected. Arbitrarily, we will place compressed n-bit data before the K bits. The four possibilities for an (n, k) mode are shown in Fig. 7.

Line Format: Later we will discuss an automatic line-to-line switching criterion for the selection of Split-Pixel modes (where $n + k$ is fixed). Thus, we introduce an additional 4-bits of identification, called SP, at the beginning of each line to specify the selected Split-Pixel mode.

Also, since the Basic Compressor operates on differences between adjacent pixels, a reference sample is required in order to be able to reconstruct actual brightness levels (in the case where Backup PCM never occurs). We are assuming a reasonably clean telecommunication channel for the Grand Tour mission(s) ($\approx 10^{-5}$ bit error rate) so that a single reference sample at the beginning of each line should be sufficient. An end-of-line identifier (e.g., a *pn* sequence) will also be provided.

CODE	BLOCK FORMAT
BACK-UP-PCM	PCM SAMPLES ID {(1) (2) (3) - - - (21)} 21 (n + k)
CODE $\overline{FS}$	K- BITS ID {CODE $\overline{FS}$} {(1) (2) (3) - - - (21)} VARIABLE 21k
FS	ID {FUNDAMENTAL SEQUENCE} {K-BITS} VARIABLE 21k
CODE FS	ID {CODE FS} {K-BITS} VARIABLE 21k

Fig. 7. Block formats.

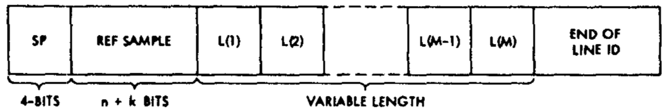

Fig. 8. Line format.

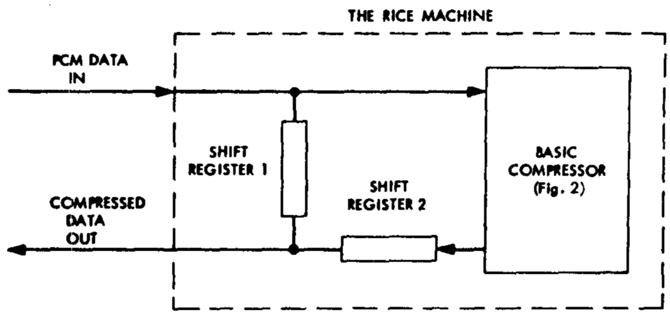

Fig. 9. Rice Machine.

Assume there are M blocks of 21 pixels in a line plus one reference sample. Let $L(i)$ be the sequence of bits resulting for the ith block. Each such $L(i)$ will be in the form of one of the formats given in Fig. 7. The resulting line format is shown in Fig. 8.

Block Diagram

A block diagram of the complete compressor is shown as the Rice Machine in Fig. 9. From the diagram, we see that the implementation of the Split-Pixel and Backup-PCM options is primarily accomplished with the addition of two shift registers. Not shown is the decision logic necessary to implement both an automatic line to line Split-Pixel mode selection (to be discussed later) and the desired data formatting.

Shift Register 1 stores the incoming PCM data until Shift Register 2 decides whether Backup PCM is to be used. The proper sequence of bits, Backup PCM or K bits, is then shifted out according to one of the formats in Fig. 7.

Performance

To see why the Split-Pixel scheme yields improved performance at high entropy levels, we need some more definitions. Let H_i denote the entropy computed for the most significant i bits and L_i denote the expected performance of the Basic Compressor on the i most significant bits.

Extensive tests indicated the following: once H_i exceeds approximately 4 bits/pixel for some i, then

$$H_i \approx H_{i-1} + 1\ . \tag{11}$$

That is, one additional bit of quantization adds approximately 1 bit to the entropy of an active picture. This is a direct consequence of the fact that each additional bit of quantization appears almost totally random.

A close look at (11) suggests a way of improving performance when H_8 is greater than 4 bits/pixel. For the sake of an explicit example suppose $H_8 = 5.5$ bits/pixel. We can use (11) twice yielding

$$H_8 \approx H_6 + 2 \tag{12}$$

or $H_6 \approx 3.5$ bits/pixel. Using the Basic Compressor on all 8 bits is inefficient since H_8 is outside of the Principal Operating Range. That is $L_8 - H_8$ is large. However, in the same way, we know that $L_6 - H_6$ is small since H_6 does lies within the Principal Operating Range. Substituting from (12) we see that

$$L_6 - H_6 \approx (L_6 + 2) - H_8. \tag{13}$$

But $L_6 + 2$ is the expected performance of a coding system which encodes the most significant 6 bits with the Basic Compressor and transmits the least significant two bits separately. We recognize this as a (6, 2) Split-Pixel mode. Performance has been improved because the Basic Compressor operates only on that part of the data for which efficient performance can be achieved.

This somewhat qualitative example should illustrate why the various Split-Pixel modes can yield improved performance at high entropy values. The graphs in Fig. 10 quantifies these results. The figure shows expected performance for the various Split-Pixel modes (given by $L_n + k$), where $n + k = 8$, as a function of 8-bit data entropy H_8. The graphs also include the effect of Backup PCM.

The (8, 0) curve is really an extension of the performance curve of Fig. 3 to high entropy values. We see that this curve rapidly pulls away from the entropy line as activity increases beyond 4 bits/pixel. This clearly indicates the need for additional modes for any mission in which very high entropy pictures are possible.

All curves level off asymptotically to 8.1 bits/pixel (the dotted portions indicate extrapolation) as a result of Backup PCM. Note that at least one of the curves is always within 0.2 bit/pixel of the entropy line (although we have extrapolated each curve for values of H_8 past 7 bit/pixel). Efficient performance is always ensured providing we can find an effective means of "almost always" selecting the best mode.

Observe that the (3, 5) curve is never best (except

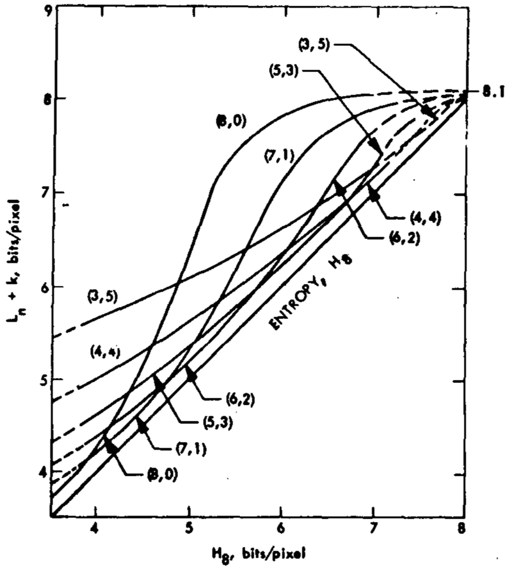

Fig. 10. Split-pixel performance.

possibly when H_8 approaches 8 bits/pixel). It was for this reason that a (3, 5) Split-Pixel mode was not included as part of the Rice Machine definition.

Line-to-Line Mode Selection

Ideally, each mode of the set where $n + k = 8$ ((8, 0), (7, 1), (6, 2), (5, 3), and (4, 4)) could be tried on, say, a line of data, and then that mode which yielded the least number of bits would be selected for actual transmission. Practically speaking this is impossible so that we need some other way of selecting modes.

Adjacent Mode Switching: Observe from Fig. 10 that as entropy increases from 3.5 bit/pixel out to 8 bit/pixel, the best mode changes in the following manner: (8, 0), (7, 1), (6, 2), (5, 3), and (4, 4). This ordering establishes an adjacency between modes. We say that the (8, 0) mode is adjacent to the (7, 1) mode, the (7, 1) mode is adjacent to both the (8, 0) mode and the (6, 2) mode, and so on. A relatively trivial mode selection decision criterion results if we allow switching between adjacent modes only. This subject was treated more fully in [5] and the results (applicable to all modes) are shown below as Table V. The criterion basically takes advantage of the fact that adjacent lines are generally highly correlated in their average properties (e.g., data activity).

L_n is interpreted here to mean the output of the Basic Compressor in bits/pixel *averaged over a complete line* where the contribution of a Backup-PCM block to the calculation of L_n is $n + 0.1$ bit/pixel.

If the indicated next mode in Table V is not one of the allowable modes this is assumed to mean "stay in the same mode."

Table V is quite consistent with earlier explanations of why the Split-Pixel modes should improve performance. When L_n exceeds 4 bits/pixel, the Basic Compressor starts to degrade in efficiency, whereas when L_n drops below 3 bits/pixel the Basic Compressor may not be working on enough of the data.

TABLE V
ADJACENT MODE SELECTOR CRITERION

Present Mode = (n, k)	Mode for Next Line
$L_n < 3$	$(n + 1, k - 1)$
$3 \leq L_n < 4$	(n, k)
$L_n \geq 4$	$(n - 1, k + 1)$

Whereas the four codes (Code $\overline{FS}$, FS, Code FS, and Backup PCM) which operate on the block level are capable of adapting to rapid changes in activity, the Split-Pixel options with adjacent mode selection provides additional capability to adapt to relatively slow but more dramatic changes in activity.

V. Using Previous Line

Introduction

This section takes the first step towards removing the initial constraint of one-dimensional processing. To keep within the realm of practicality, we assume here only that a "very-long" shift register is available for storing one line of PCM data. To take advantage of the line to line correlation we make use of the more general coding system of Fig. 1.

We first specialize the notation of Fig. 1 to television by defining $a_{i,j}$ as the jth sample in the ith line. The developments of Sections II–IV were motivated by the conclusion that, of those samples preceding $a_{i,j}$ in the ith television line, only $a_{i,j-1}$ contributed significantly to the prediction of $a_{i,j}$. Similarly, we conclude that of all samples in the preceding line, only $a_{i-1,j}$ (the sample immediately above $a_{i,j}$) contributes significantly to the prediction of $a_{i,j}$. From the symmetry in image data, the best linear mean-square estimate (with truncation) of $a_{i,j}$ based on both $a_{i,j-1}$ and $a_{i-1,j}$ can be written as

$$\hat{a}_{i,j} = \left\lfloor \frac{a_{i,j-1} + a_{i-1,j}}{2} \right\rfloor \tag{14}$$

where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. We then apply the Rice Machine to the coding of the differences

$$d_{i,j} = a_{i,j} - \hat{a}_{i,j} \tag{15}$$

Computing this difference amounts to an insignificant increase in operational complexity. Thus the only real hardware cost in using this estimate is the memory required for storing a single line of data.

Performance

As in the one-dimensional case we first investigate the distributions of $d_{i,j}$ given by (14) and (15). We note that

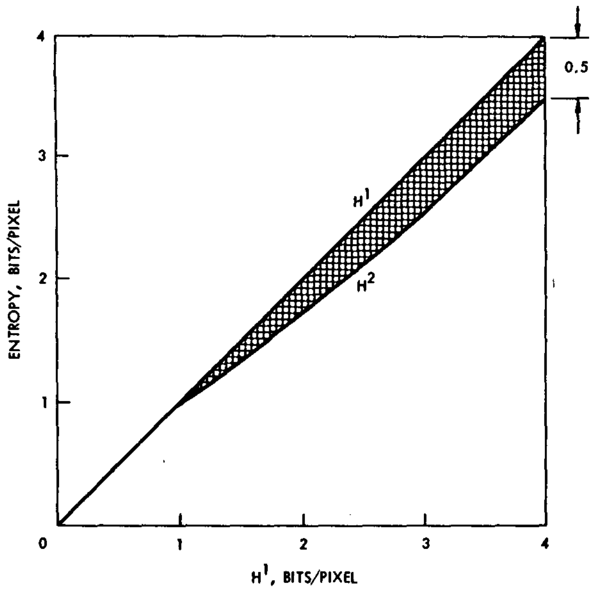

Fig. 11. H^2 versus H^1.

the resulting distributions have precisely the same form as those observed for the one-dimensional case. The best *a priori* ordering of differences is still supplied by condition (1). Thus the set of typical distributions is the same in both cases.

Since the performance of the Rice Machine is specified for these distributions by the $E[L(P)]$ versus $H(P)$ curves of Figs. 3 and 10 we need only investigate the correspondence between the 1-dimensional and 2-dimensional entropies for the same uniform data.

This comparison has been plotted for the Principal Operating Range in Fig. 11 with H^1 denoting the entropy of differences when $d_{i,j} = a_{i,j} - a_{i,j-1}$ and H^2 denoting the entropy of differences when $d_{i,j}$ is given by (14) and (15).

We note that $H^1 \approx H^2$ in the range $0 \leq H^1 \leq 1$. Beyond this point $H^1 - H^2$ increases to a constant value of approximately 0.5 bit/pixel at $H^1 \approx 3$.

Noting that condition (11) holds also for H^2 is enough to show that $H^1 - H^2 \approx 0.5$ for values of $H^1 > 4$. This checks with experimental data.

Thus we can say that the estimate in (14) and (15) improves performance by about 0.5 bit/pixel for data with 1-dimensional difference entropies exceeding 3 bits/pixel. The improvements are somewhat smaller for less active data.

References

[1] JPL Staff, "The thermoelectric outer planet spacecraft (TOPS) mission," *Astronaut. Aeronaut.*, Sept. 1970.

[2] R. F. Rice, "The code word wiggle: TV data compression," Jet Propulsion Lab., California Inst. Technol., Pasadena, Tech. Memo. 33-428, June 15, 1969.

[3] ——, "The code word wiggle: Implementation," Jet Propulsion Lab., California Inst. Technol., Pasadena, JPL Internal Doc. 900-273, Rev. A, June 1970.

[4] ——, "The vertical wiggle: Definition and implementation," Jet Propulsion Lab., California Inst. Technol., JPL Internal Doc. 900-304, Rev. A, July 1970.

[5] ——, "The Rice machine," Jet Propulsion Lab., California Inst. Technol., JPL Internal Doc. 900-408, Sept. 1, 1970.

[6] ——, "TV data compression for a Grand Tour of the outer planets," presented at 4th Annu. Hawaii Int. Conf. System Sciences, January 1971.

[7] H. Abramson, *Information Theory and Coding.* New York: McGraw-Hill, 1963.

[8] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

Robert F. Rice received the B.S. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, N. Y., and the M.S. degree from the University of California, Los Angeles, where he is also pursuing the Ph.D. degree in communication systems.

In 1966 he joined the Jet Propulsion Laboratory, California Institute of Technology, Pasadená, where he became a Senior Research Engineer working on spacecraft data compression systems.

Mr. Rice is a member of Tau Beta Pi and Eta Kappa Nu.

James R. Plaunt was born in Ames, Iowa, on May 14, 1947. He received the B.A. degree in mathematics from Claremont Men's College, Claremont, Calif., in 1969. He is working toward the Ph.D. degree in applied mathematics at the University of California, Los Angeles.

He joined the Jet Propulsion Laboratory, Pasadena, where he has been engaged in research on spacecraft image data processing and compression.

Mr. Plaunt is a member of Pi Mu Epsilon.