NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-428

The Code Word Wiggle: TV Data Compression

R. F. Rice

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

June 15, 1969

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-428

The Code Word Wiggle: TV Data Compression

R. F. Rice

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

June 15, 1969

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

Preface

The work described in this report was performed by the Space Sciences Division of the Jet Propulsion Laboratory.

Acknowledgment

The IBM System 360/44 programming used herein was done by James Plaunt of Claremont Men's College.

Contents

Contents (contd)

Tables

Figures

Abstract

The planned exploration of the outer planets has generated the need for information-preserving, data-compression systems that are applicable to spacecraft scientific imaging experiments. A variable-length coding algorithm that is capable of adapting to the nonstationary nature of digital spacecraft data has been developed for noiseless channel conditions. Performance equations are first developed for a suitable analytic model, and then applied directly to 6-bit, pulse-code-modulated TV data. The adaptive algorithm adjusts to changes in picture activity by treating blocks of consecutive picture elements as individual data sources. Using a single 8-word binary code, output data rates remain close to element-to-element entropy values for each block (and therefore for each picture). The preliminary dynamic performance curves (average bits per pixel vs entropy per data block) are shown.

The Code Word Wiggle: TV Data Compression

I. Introduction

The motivation for this report was generated by the need for an information-preserving, data-compression system applicable to spacecraft scientific imaging experiments. By *information preserving* we mean that the output of a TV pulse-code-modulated (PCM) system can be completely reconstructed from the compressed data. A study of various bandwidth-compression schemes by Anderson (Ref. 1) concluded that, at least for the present, emphasis should be placed on such systems. Some work in other areas is continuing at JPL (Kleinrock, Ref. 2).

An early study by Electro-Mechanical Research, Inc. (Ref. 3) investigated a rather simple form of variable-length coding. The compression results were not significant due to the restricted form of the coding scheme. Here we present a variable-length coding algorithm which is capable of adapting to the nonstationary nature of digital spacecraft TV data. It should be noted that all discussion and development are limited to noiseless channel conditions.

Briefly, the coding system derives its adaptive character by the following operations. Each line is divided into blocks of consecutive picture elements and each of these blocks is treated as an individual data source. Each data block is coded according to the algorithm developed in Section II. This algorithm provides "three codes in one" (i.e., only a single 8-word binary code is stored) and automatically generates its own decision mechanism for selection of the best code for each block. In this manner output data rates remain close to the element-to-element entropy values for each block (and therefore for each picture).

The terminology and performance equations for the basic algorithm (the method for coding each data block) are developed in a general context in Section II. Particular orientation to the TV application is avoided until Section III, where these results are specialized and extended to 6-bit PCM *Surveyor* and *Ranger* pictures. Selected pictures, exhibiting a wide range of activity, are used to validate the derived performance equations by actual coding. An adaptive system is specified and preliminary dynamic performance curv :s (average bits per pixel vs entropy per data block) are shown.

II. Coding Algorithm

We first consider a transformation by which a general zero memory source may be efficiently encoded as the nth extension of an equivalent binary source.

A. Fundamental Sequence

We interpret our information source as a source which emits a sequence of q symbols from a fixed finite alphabet $S = \{s_1, s_2, s_3, \cdots, s_q\}$. For the special case of a zero memory source, successive symbols are statistically independent and we denote their probability of occurrence by the distribution

$$Pr[s_i] = p_i, i = 1, 2, \cdots, q \tag{1}$$

where

$$p_1 \geq p_2 \geq p_3 \geq \cdots \geq p_q \tag{2}$$

We define the mean of the input source as follows:

$$E(S) = \sum_{i=1}^{q} i p_i \tag{3}$$

The entropy of this source is denoted by $H(S)$, where

$$H(S) = \sum_{i=1}^{q} p_i \log \frac{1}{p_i} \tag{4}$$

From this model the first problem is to convert a sequence of these q symbols into an equivalent binary sequence.

1. Sample matrix. Consider a block of J data samples z_i (e.g., a sequence of J consecutive TV picture elements). We can designate this sequence by

$$z_1 z_2 \cdots z_J \tag{5}$$

Construct the $q \times J$ sample matrix with element a_{ij} given by

$$a_{ij} = \begin{cases} 1 & \text{iff } z_j = s_i \\ 0 & \text{otherwise} \end{cases} \quad \begin{matrix} i = 1, 2, \cdots, q \\ j = 1, 2, \cdots, J \end{matrix} \tag{6}$$

This is illustrated in Fig. 1 for $q = 4$, $J = 8$, and an assumed input sequence $s_1, s_1, s_2, s_2, s_1, s_3, s_4, s_1$.

2. Wiggle operation. Note that the initial input sequence is completely specified by the location of the ones in this matrix. In fact, the matrix of Fig. 1 is completely specified by the first $q - 1$ rows. One obvious binary sequence which is equivalent to the original sequence is formed by row 1 followed by row 2, row 3, etc. This is of course very inefficient.

Observe that a single "1" appears in each column. Hence, the first appearance of a "1" completely specifies the column or position in which it appears for all subsequent rows. This is indicated in Fig. 2 for the example by crossing out all zeros located below a "1" in a previous row. The last row may be crossed out under any circumstance since it is uniquely specified by the knowledge of all preceding rows.

A new binary sequence which completely defines the original input sequence is generated by the remaining elements of rows 1 through $q - 1$. This sequence is traced out for the example in Fig. 2. We call this the

Sample	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
Assumed input sequence	s_1	s_1	s_2	s_2	s_1	s_3	s_4	s_1
s_1	1	1	0	0	1	0	0	1
	0	0	1	1	0	0	0	0
s_3	0	0	0	0	0	1	0	0
s_4	0	0	0	0	0	0	1	0

Fig. 1. Sample matrix

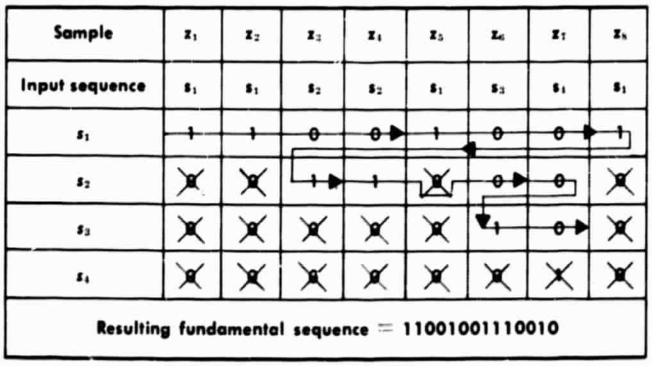

Fig. 2. Fundamental sequence

"wiggle operation" and define the resulting sequence as the *fundamental sequence* (FS). It is this sequence we propose to code.

B. Length of the Fundamental Sequence

To find the length of the FS we will consider the contribution of each column (Fig. 2).

Let $\tilde{d}_j$ denote the contribution of the jth column to the FS, where $j = 1, 2, \cdots, J$. By the construction algorithm, if a "1" appears in the ith row ($i = 1, 2, \cdots, q - 1$), then all remaining zeros in that column are deleted from further consideration. Hence, the contribution is precisely i bits. Thus the random variable $\tilde{d}_j$ takes on the value i. If a "1" appears in the qth row, then $\tilde{d}_j = q - 1$. Hence, we have the result

$$Pr[\tilde{d}_j = i] = Pr[z_j = s_i] = p_i \; \forall i \leq q - 2$$
$$Pr[\tilde{d}_j = q - 1] = p_{q-1} + p_q \tag{7}$$

The length of the fundamental sequence F can now be written as

$$F = \sum_{j=1}^{J} \tilde{d}_j \tag{8}$$

where the $\tilde{d}_j$ are seen to be independent random variables all distributed as in Eq. (7). The mean of F is given by

$$E(F) = J[E(S) - p_q] \tag{9}$$

But by the law of large numbers, F converges in probability to its mean. Hence, for J sufficiently large we have

$$F = E(F) = J[E(S) - p_q] \tag{10}$$

In words, Eq. (10) means that *if the fundamental sequence is transmitted, the number of bits per input sample approximately equals the mean of the input distribution.* It gives some quantitative meaning to the ordering of probabilities given in (2). Later we will see that this result has another useful interpretation.

C. Coding the Fundamental Sequence

The coding scheme employed simply involves coding the nth extension of the binary source represented by the fundamental sequence.

As an example, consider the fundamental sequence derived in Fig. 2.

11001001110010

Coding the second extension implies coding pairs as shown below:

(11)(00)(10)(01)(11)(00)(10)

If the third extension is coded, a dummy zero must be inserted because 3 does not divide the length of the fundamental sequence (i.e., 14 bits here).

(110)(010)(011)(100)(100) ← dummy

To specify the general case, let Z_i^n denote the ith n-bit binary sequence composing the FS (there are F/n such sequences). Then the FS will appear as follows:

$$FS = Z_1^n \, Z_2^n \cdots Z_{F/n}^n \tag{11}$$

The reader should compare this with the original input sequences given by (5).

1. Code set. Coding the nth extension of a binary source requires 2^n code words. Denote by $\{\ell_i\}$, $i = 1, 2, \cdots, 2^n$ a set of variable-length code words which can be derived by the Huffman coding algorithm for some distribution (Ref. 4). The symbol ℓ_i takes on a dual meaning. It refers not only to the ith code word in the set $\{\ell_i\}$ but also to the length of that word in bits. Denote such a code by Code $(\ell_1, \ell_2, \cdots, \ell_{2^n})$. We will use this notation extensively in Section III. Furthermore, we define a code to be *linear* if $\ell_i = i \; \forall i < 2^n$ and $\ell_{2^n} = 2^n - 1$.

For the present discussion assume that the words of an arbitrary code are labeled such that

$$\ell_1 \leq \ell_2 \leq \ell_3 \leq \cdots \leq \ell_{2^n} \tag{12}$$

If we assume the probability of a "1" is less than or equal to 1/2, the n-bit sequences $\{S_i\}$ (and corresponding code words) may be ordered according to decreasing probabilities, as shown in Table 1.

Define $\{L(j)\}$ as the set of all code words corresponding to n-bit binary sequences with j ones. The dual meaning here implies that the term $L(j)$ takes on a value equal to

Table 1. Code word ordering

Number of ones in n-bit sequence	Sequence number s_i	Code word ℓ_i	L(j)
0	s_1	ℓ_1	L(0)
1	s_2	ℓ_2	
•	•	•	
•	•	•	L(1)
•	•	•	
1	$s_{1+\binom{n}{1}}$	$\ell_{1+\binom{n}{1}}$	
2	$s_{1+\binom{n}{1}+1}$	$\ell_{1+\binom{n}{1}+1}$	
•	•	•	
•	•	•	L(2)
•	•	•	
2	$s_{1+\binom{n}{1}+\binom{n}{2}}$	$\ell_{1+\binom{n}{1}+\binom{n}{2}}$	
•	•	•	•
•	•	•	•
•	•	•	•
n	s_{2^n}	ℓ_{2^n}	L(n)

the sum of the lengths of all code words in $\{L(j)\}$. Specifically,

$$L(j) = \sum_{i=n_1}^{n_2} \ell_i \tag{13}$$

where

$$n_1 = 1 + \sum_{k=0}^{j-1} \binom{n}{k} \qquad n_2 = \sum_{k=0}^{j} \binom{n}{k}$$

2. ***Block diagram.*** A system block diagram is shown in Fig. 3.

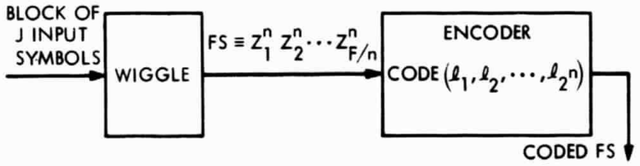

Fig. 3. System block diagram

D. Analysis

In this paragraph we will derive an equation which specifies the performance of this system for an arbitrary code and from which an optimum set of code words can be derived. The analysis[1] will proceed by treating each row of Fig. 2, along the path of the wiggle operation, as a separate source. We first require some additional definitions.

1. ***Parameters.***

Let

$$\tilde{J}_m = \{\text{the number of columns or samples remaining after symbol } s_{m-1} \text{ (row } m-1) \text{ has been analyzed}\} \tag{14}$$

For instance, in the example of Fig. 2 we have

$$(\tilde{J}_1 = 8, \tilde{J}_2 = 4, \tilde{J}_3 = 2, \tilde{J}_4 = 1)$$

Let A_{i_m} denote the i_mth column of the set of $\tilde{J}_m$ remaining columns, where

$$i_m = 1, 2, \cdots, \tilde{J}_m \tag{15}$$

and define

$$\tilde{x}_{i_m} = \begin{cases} 1 & \text{If } s_m \in A_{i_m} \text{ under the condition } s_1 \cup s_2 \cup \cdots \cup s_{m-1} \notin A_{i_m} \text{ (with probability } \Pi_m) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

As an example here, consider the second row (s_2) of Fig. 2. We have

$$(A_1 = \text{col. } 3, A_2 = \text{col. } 4, A_3 = \text{col. } 6, A_4 = \text{col. } 7)$$

and

$$(\tilde{x}_1 = 1, \tilde{x}_2 = 1, \tilde{x}_3 = 0, \tilde{x}_4 = 0)$$

Making use of these definitions yields

$$\Pi_m = Pr[\tilde{x}_{i_m} = 1 \,|\, s_1 \cup s_2 \cup \cdots \cup s_{m-1} \notin A_{i_m}]$$

$$= \frac{Pr[s_m \cap (s_m \cup s_{m+1} \cup \cdots \cup s_q)]}{Pr[s_m \cup s_{m+1} \cup \cdots \cup s_q]}$$

[1]A similar analysis appeared in Ref. 5.

But the events $\{s_i\}$ are mutually exclusive, therefore

$$\Pi_m = \frac{p_m}{\sum_{i=m}^{q} p_i} \tag{17}$$

We shall also require the distribution for $\tilde{J}_m$.

$$Pr[\tilde{J}_m = k] = Pr[\text{Any of the symbols } s_1, s_2, \cdots, s_{m-1} \text{ have appeared in any of } J-k \text{ of the } J \text{ columns}] \tag{18}$$

where $k = 0, 1, 2, \cdots, J$

But the probability that any position has one of the symbols $s_1, s_2, \cdots, s_{m-1}$ present is given by

$$Pr\,[s_1 \cup s_2 \cup \cdots \cup s_{m-1}] = \sum_{i=1}^{m-1} p_i = \zeta_m \tag{19}$$

Then Eq. (18) becomes

$$Pr[\tilde{J}_m = k] = \binom{J}{J-k} (\zeta_m)^{J-k} (1 - \zeta_m)^k = \lambda_{mk} \tag{20}$$

2. ***Performance.*** If we let $\tilde{T}_m$ denote the total number of bits required for s_m, then the expected number of bits per sample for a block size of J is given by

$$\bar{L} = \frac{1}{J} \sum_{m=1}^{q-1} E[\tilde{T}_m] \tag{21}$$

The remaining problem is to evaluate $E[\tilde{T}_m]$, $m = 1, 2, \cdots, q-1$ for the prescribed coding scheme.

Let $\tilde{T}_m(k)$ define the number of bits transmitted for s_m given that $\tilde{J}_m = k$.

Then[2]

$$E[\tilde{T}_m(k)] = \frac{k}{n} \sum_{j=0}^{n} \Pi_m^j (1 - \Pi_m)^{n-j} \cdot L(j) \tag{22}$$

[2]Actually this equation is inexact if n does not divide k. However, these error terms are negligible for $J \gg n$ and any practical input distribution.

Multiplying by λ_{mk} and summing on k, we get

$$E[\tilde{T}_m] = \frac{1}{n} \sum_{j=0}^{n} \Pi_m^j (1 - \Pi_m)^{n-j} \left[\sum_{k=0}^{J} k\lambda_{mk} \right] L(j)$$

Noting that $\tilde{J}_m$ is binomial with parameter $(1 - \zeta_m)$ we get—see Eq. (21)—for the average bits per input sample

$$\bar{L} = \sum_{j=0}^{n} L(j) \left[\frac{1}{n} \sum_{m=1}^{q-1} (1 - \zeta_m) \Pi_m^j (1 - \Pi_m)^{n-j} \right] \tag{23}$$

Substitution of $L(j)$ from Eq. (13) leads to the form

$$\bar{L} = \sum_{i=1}^{2^n} P_i \ell_i \tag{24}$$

from which an optimum set of code words can be derived such that $\bar{L}$ is minimized (Refs. 4 and 6).

3. ***Fundamental sequence.*** If the lengths of all 2^n code words are set equal to n-bits then Eq. (23) defines the average bits per sample required if the fundamental sequence is transmitted. Substituting in Eq. (23) and reversing the order of summation we get

$$\frac{1}{J} E(F) = \sum_{m=1}^{q-1} (1 - \zeta_m) \left[\sum_{j=0}^{n} \binom{n}{j} \Pi_m^j (1 - \Pi_m)^{n-j} \right]$$

$$= \sum_{m=1}^{q-1} (1 - \zeta_m)$$

$$= \sum_{m=1}^{q-1} \left(1 - \sum_{i=1}^{m-1} p_i \right)$$

After suitable manipulations we get

$$E(F) = J[E(S) - p_q] \tag{25}$$

where $E(S)$ is the mean of the input source probability distribution. Note that we previously obtained this result more directly and in fact derived the stronger result given by Eq. (10).

4. ***Complementary code.*** If a source is very inactive, some of the Π_m may exceed ½. In this case the best ordering of code words as defined in Table 1 is reversed.

This can be accommodated in Eq. (23) by interchanging Π_m and $(1 - \Pi_m)$ in all instances where $\Pi_m > 1/2$. In an actual system this simply requires that the sequence of zeros and ones generated for the FS by s_m be complemented before coding. We defer the details of this principle until Section III where we will apply the complementary code to TV data by complementing an entire FS.

E. The Wiggle Squared

One might wonder what happens when the wiggle operation is performed a second time (i.e., on the fundamental sequence). Such an operation is indicated in Fig. 4.

The input to the second box is the fundamental sequence $FS \equiv Z_1^n Z_2^n \cdots Z_{F/n}^n$ formed by F/n symbols from the alphabet $\{S_1, S_2, \cdots, S_{2^n}\}$ where the S_i are defined in Table 1.

The sample matrix for the second box is a $(2^n \times F/n)$ matrix with binary entries $\{a_{ij}\}$ determined as follows:

$$a_{ij} = \begin{cases} 1 & \text{iff } Z_j^n = S_i \\ 0 & \text{otherwise} \end{cases} \qquad \begin{aligned} i &= 1, 2, \cdots, 2^n \\ j &= 1, 2, \cdots, F/n \end{aligned} \tag{26}$$

The fundamental sequence for this sample matrix is derived exactly as in Fig. 2. We denote this output sequence by $(FS)^2$ and its length by FF.

To find the average length of the $(FS)^2$ we use an argument completely analogous to the development for the fundamental sequence itself; see Eqs. (7) and (8). First let

$$P_{i,j}^n = Pr\,[Z_j^n = S_i] \tag{27}$$

where

$$i = 1, 2, \cdots, 2^n;\; j = 1, 2, \cdots, F/n$$

we have

$$E(FF) = \sum_{j=1}^{F/n} [E(Z_j^n) - P_{2^n,j}] \tag{28}$$

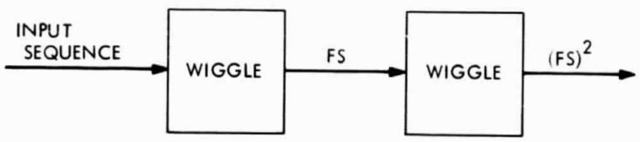

Fig. 4. Wiggle squared

Consider a single term

$$E(Z_j^n) - P_{2^n,j} = \sum_{i=1}^{2^n-1} iP_{i,j} + (2^n - 1)\,P_{2^n,j}$$

But the right-hand side equals the mean number of bits transmitted if the jth sequence Z_j^n is coded by the linear code, Code$(1, 2, 3, \cdots, 2^n - 1, 2^n - 1)$. Since this is true for all j, we have the result:

In terms of expected bits per input sample, the wiggle squared operation is equivalent to coding the fundamental sequence with the linear code.

Quantitatively, we have

$$E(FF) = J\bar{L} \tag{29}$$

where $\bar{L}$ is given by Eqs. (23) and (24) with appropriate substitutions for the linear code.

Note that a similar interpretation can be applied to the FS and coding the input source with the linear code, Code$(1, 2, \cdots, q - 1, q - 1)$.

III. Application to TV

In this section the coding algorithm is applied to 6-bit *Ranger* and *Surveyor* pictures. To suitably test the basic algorithm, seven pictures were selected having a wide range in data activity. Reproductions of these pictures are shown in Fig. 5 in order of increasing entropy values H^*. All further reference to sample pictures will use this numbering system. A nomenclature table of terms frequently used in this section is in the rear of this memorandum.

A. Data Model

To apply the coding algorithm to TV data, specific parameters must be identified with the more general terms of Section II.

Let b_i represent the ith quantum level of the TV PCM system, where $i = 1, 2, \cdots, 64$. Suppose, for example, that the distribution of the $\{b_i\}$ is known for a particular section of a picture. For the sake of simplicity assume

$$Pr[b_1] \geq Pr[b_2] \geq \cdots \geq Pr[b_{64}] \tag{30}$$

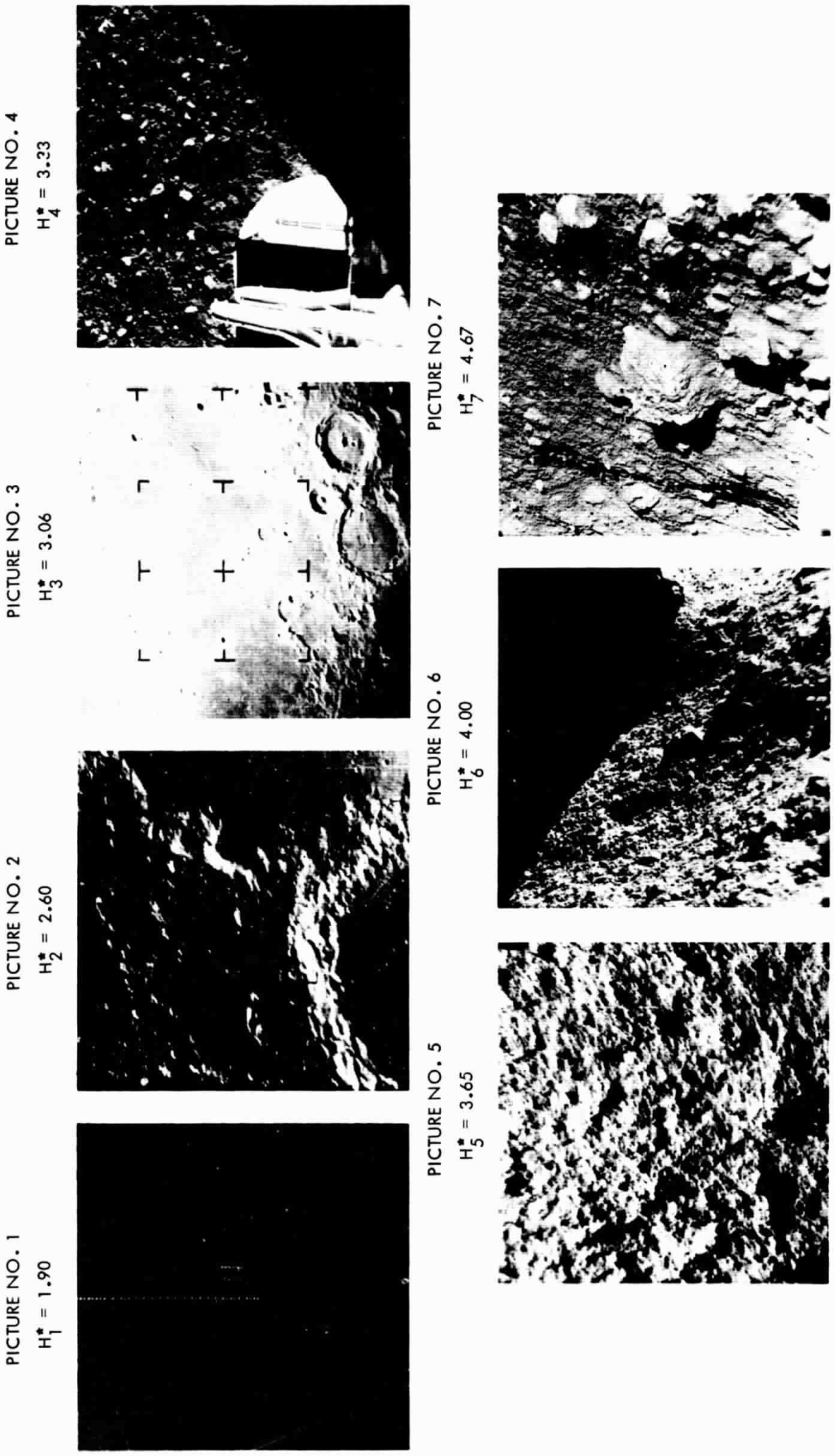

Fig. 5. Sample pictures

Then condition (2) could be satisfied by identifying b_i with s_i. By Eq. (10), a fundamental sequence generated in this section would be minimized; however, this selection of $\{s_i\}$ could be bad for another section. Hence we need another means of selecting the input symbols, $\{s_i\}$.

1. Selection of $\{s_i\}$. One model which consistently provides a good approximation to actual output conditions is summarized by the following: given that quantum level b_i has just occurred, the distribution of quantum levels for the next sample point is unimodal about b_i. Quantitatively we have

$$Pr[b_i \mid b_i] \geq Pr[b_{i+1} \mid b_i] \geq Pr[b_{i-1} \mid b_i] \geq Pr[b_{i+2} \mid b_i] \geq \cdots \quad (31)$$

Since we make the same assumption for all levels, (31) can be written in terms of first-order differences. Define $\tilde{d}$ as the difference between the present and previous sample. Then $\tilde{d}$ takes on values between -63 and $+63$. The inequalities in (31) become

$$Pr[\tilde{d} = 0] \geq Pr[\tilde{d} = +1] \geq Pr[\tilde{d} = -1] \geq Pr[\tilde{d} = +2] \geq \cdots \quad (32)$$

Condition (2) is satisfied if we associate the symbols $\{s_i\}$ with these first-order differences in the order specified by Eq. (32). Specifically

$$\begin{array}{ccc} s_1 & \longleftrightarrow & \tilde{d} = 0 \\ s_2 & \longleftrightarrow & \tilde{d} = +1 \\ s_3 & \longleftrightarrow & \tilde{d} = -1 \\ s_4 & \longleftrightarrow & \tilde{d} = +2 \\ \vdots & \vdots & \vdots \\ s_{127} & \longleftrightarrow & \tilde{d} = -63 \end{array} \quad (33)$$

This assignment will be assumed in all further discussions.

2. Bounding performance. If we could completely define the statistics for a data source then theoretically we could bound the performance of any coding system by the entropy of the source. Unfortunately, the source we are considering is a class of TV pictures which taken even individually can be highly nonstationary. Hence we must make some approximations to get a handle on the problem.

By definition, the basic coding algorithm operates on blocks of J consecutive picture elements. We can associate with each block (1) a unique positive integer n, (2) an entropy $H(n)$, and (3) an average word length (bits per pixel) $L(n)$.[3]

If there are N blocks of J picture elements, then we have

$$H(n) \leq L(n) \qquad n = 1, 2, \cdots, N \quad (34)$$

We define the entropy of the picture by

$$\hat{H} = \frac{1}{N} \sum_{n=1}^{N} H(n) \quad (35)$$

and the average word length for the picture by

$$\hat{L} = \frac{1}{N} \sum_{n=1}^{N} L(n) \quad (36)$$

From the inequality in (34) we note that

$$\hat{H} \leq \hat{L} \quad (37)$$

To obtain $H(n)$ we note that (1) if the nth block of TV data is essentially first-order Markov (next quantum level depends only on the present quantum level), and (2) if the conditional entropies are approximately the same for those quantum levels which occur frequently in the nth block, then the entropy calculated from the distribution of first differences in (32) is a good approximation to $H(n)$.

Both conditions (1) and (2) are good assumptions if the picture activity within a block is uniform. Certainly the latter condition will exist more frequently if the block size is small.

Henceforth, all entropy calculations will be derived from distributions of differences. We understand that any $H(n)$ computed in this way probably is a good but not best lower bound to the coding capability for the nth block.

We define a *basic system* as one which performs the wiggle algorithm with a single code according to the assignment in (33). The performance for this system is bounded by the entropy computed from the distribution

[3]This should not be confused with $L(j)$ given previously in Eq. (13).

of differences for the whole picture H^*. (These are the entropy values appearing in Fig. 5.) This approximates $\hat{H}$ if the picture does not have drastic changes in activity, i.e., the $H(n)$ do not fluctuate wildly. The basic system is equivalent to treating the whole picture as a single block of data.

We will first present the results for the basic system and then demonstrate how the system can be modified to take full advantage of drastic scene changes within a given picture.

B. Performance of the Basic System

1. Validation of analysis. The analysis of Section II was validated by comparing predicted performance for the second extension with actual results obtained by applying the algorithm directly. Predicted and actual performance for the 4-word binary code, Code(1233), are shown in Fig. 6. The predicted curve was obtained by substituting histograms of first-order differences into Eq. (23) according to the assignment given in (33). A similar comparison for the fundamental sequence in Eq. (10) was exact and is therefore shown as a single curve. These results should instill confidence in Section II. Hence all further discussion will be based on computed results only.

2. Single code performance. Calculations will be limited to the second and third extensions. As will become clear, it appears that little is to be gained by going to higher extensions.

The selection of codes is particularly simple. For the second extension there are only two compact codes of 4-code words, Code(1233) and Code(2222). The latter is equivalent to the fundamental sequence and the former is equivalent to $(FS)^2$. Both of these have already been shown in Fig. 6. For the third extension there are 16 compact codes of 8 code words, but surprisingly Code(13335555) gives a minimum average word length for pictures 3–7 and does reasonably well on pictures 1 and 2. Therefore, we will restrict our attention to this code.

Performance curves for Code(1233) and Code(13335555) are shown in Fig. 7 along with plots for the fundamental sequence and entropy of first differences H^*.

C. Improved Low End Performance

Note that the performance curves in Fig. 7 can be considered as continuous functions of entropy (i.e., we

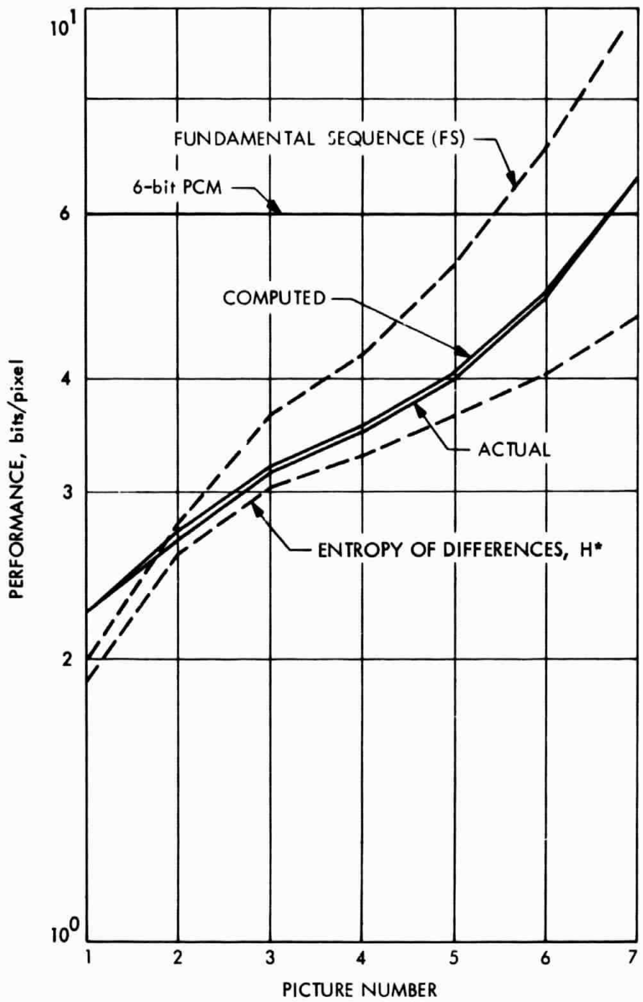

Fig. 6. Computed vs actual for Code(1233)

can extrapolate between data points for the seven sample pictures). Several other pictures with various entropy values were found to satisfy this interpretation. This allows us to drop reference to particular pictures and in fact to interpret these curves as dynamic representations of performance as activity levels change throughout a picture, e.g., a plot of $L(n)$ vs $H(n)$.[4]

By assumption we have thus far considered only systems whose performance dynamically moves along *only one* of these curves. We will now introduce a simple modification which allows a single code system to adapt to drastic changes in data activity. This is accomplished

[4]This interpretation is somewhat inexact since not all of the sample pictures are uniformly active. We will proceed with the understanding that more precise dynamic curves can be generated from selected uniform data.

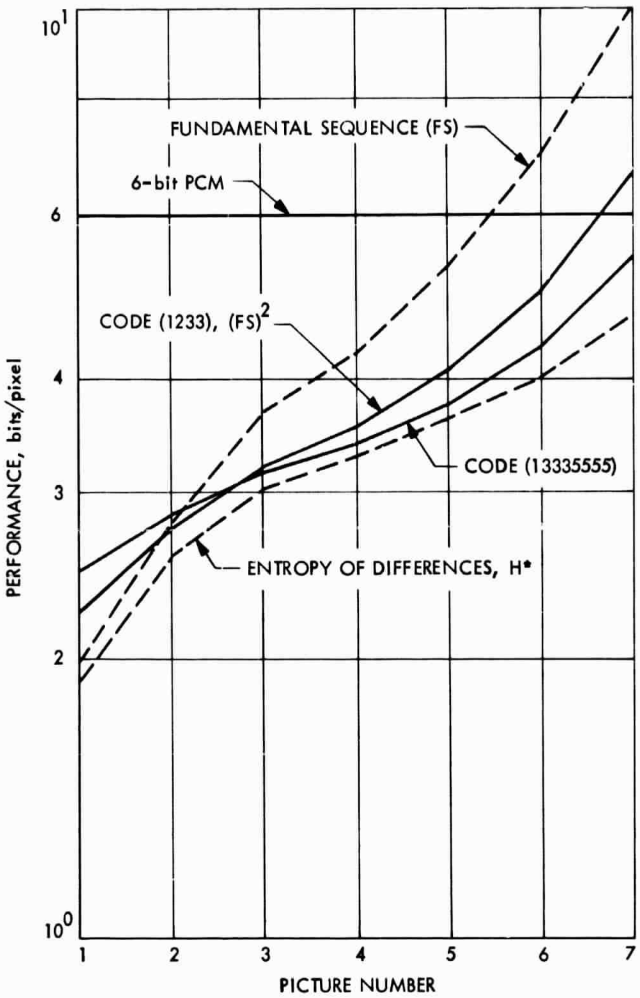

Fig. 7. Computed single code performance

with essentially no increase in functional complexity. In particular we will discuss modifications for the Code(13335555) system. It should be clear that similar modifications and conclusions can be obtained for the Code(1233) system in an analogous fashion.

1. Adding the FS. Note that the curve for the fundamental sequence in Fig. 7 crosses the Code(13335555) curve at an approximate value of 2.9 bits/pixel. For a block size of J picture elements, the length of the FS at this point is $(2.9)J$ bits. Define the threshold T_1 by

$$T_1 = 2.9 \tag{38}$$

Then, if the length of the FS is less than JT_1 bits, it is better to transmit the FS without coding.

Thus far, the modified single code system operates *dynamically on a block-by-block basis* as follows: If $F < JT_1$, the *uncoded* FS is transmitted; whereas if $F \geq JT_1$, the *coded* FS is transmitted. By this alteration, the performance of the Code(13335555) system is substantially improved in regions of inactivity.

2. Complementary code. An area of a picture that has no change in signal (e.g., a horizon scene) might come under the description of "highly inactive region." If these regions are expected to occur frequently, it becomes important to achieve a high degree of compression whenever they are encountered.[5] Note that the performance of the FS will converge to a value of 1 bit/pixel as the entropy goes to zero. This follows from the fact that all ones in the sample matrix of Fig. 1 will appear in the first row (therefore row 1 is the FS).

The 6:1 compression figure here can again be substantially improved by recalling the concept of a complementary code given in the previous section. We define the complementary code by

$$\overline{\text{Code}}(13335555) \triangleq \text{Code}(55553331) \tag{39}$$

But coding the FS with the $\overline{\text{Code}}(13335555)$ is identically equivalent to first complementing each bit of the FS and then coding with the original Code(13335555).

To see the reasoning here, again consider the limiting case when all ones of the sample matrix of Fig. 1 appear in row 1. If the original Code(13335555) is used to code the FS, this means that each subsequence of three ones will be coded into a 5-bit code word. The result is 5/3 bits/pixel. However, if (39) is used, then each subsequence of three ones is coded into a 1-bit code word with the result, 1/3 bits/pixel. It is easy to see that the same result is obtained if the FS is first complemented (i.e., row 1) and then coded with Code(13335555).

Hence we can define a second threshold for the FS, T_2, at the point where the performance curves for the FS and the $\overline{\text{Code}}(13335555)$ cross. Since actual data on highly inactive TV pictures are not presently available, we estimate a value of 1.5 bits/pixel for illustrative purposes. The transmission criterion of the combined system is shown in Table 2 and the dynamic performance curves appear in Fig. 8. The meaning of the introductory statement "three codes in one" should now be apparent.

[5]The minimum entropy achieved for such scenes depends directly on the sensor signal-to-noise ratio (S/N) and approaches zero as S/N becomes large. We will henceforth assume a "good" system to illustrate what is possible.

Table 2. Transmission criterion for the modified Code(13335555) system

Length of FS	$J \le F < JT_2$	$JT_2 \le F < JT_1$	$F \ge JT_1$
Transmission criterion	$\overline{\text{Code}}$(13335555)	FS	Code(13335555)

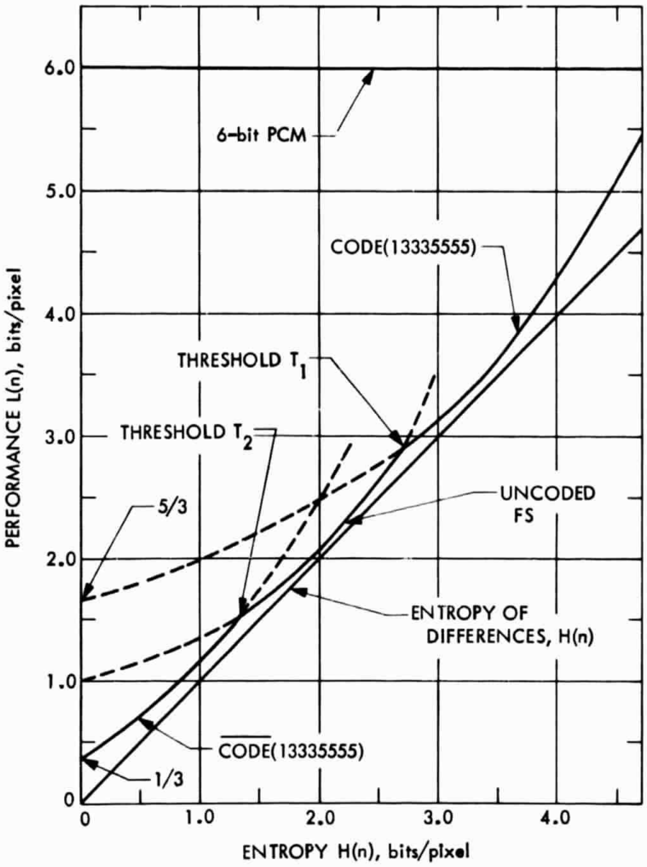

Fig. 8. Dynamic performance curves for modified Code(13335555) system

3. Example. As a crude example, consider a picture in which the top half is pure horizon (all black, $H = 0$) and the bottom half has an entropy value of 3.0 bits. We wish to compare the performance of the modified Code(13335555) system with that of the basic Code-(13335555) system.

The basic system operates along the Code(13335555) curve in Fig. 8 throughout the picture. Hence the expected number of bits per pixel becomes simply

$$L_B = \frac{1}{2}(1.67) + \frac{1}{2}(3.1) \cong 2.4 \text{ bits/pixel} \quad (40)$$

For the modified system, we note (see Fig. 8 and Table 2) that the top half of the picture operates under the $\overline{\text{Code}}$(13335555) whereas the bottom half operates under the Code(13335555). The expected number of bits per pixel becomes

$$L_M = \frac{1}{2}(0.33) + \frac{1}{2}(3.1) \cong 1.7 \text{ bits/pixel} \quad (41)$$

Thus the modified system achieves an additional reduction of 0.7 bits/pixel. Note that this compares with an overall picture entropy of 1.5 bits/pixel; see Eq. (35).

D. Discussion

From Fig. 8 we observe that the modified Code-(13335555) system performs reasonably well regardless of the activity level for a section of a picture; i.e., the output data rate $L(n)$ is close to the entropy for that section of the picture $H(n)$. This implies good performance for the whole picture [see Eqs. (35) and (36)]. This point was made evident by the previous example. In fact we can say that for the class of pictures considered here, Fig. 8 predicts good performance independent of the particular picture. Thus applicability of this system includes planetary flybys, orbiters, and landers.

Figure 8 and the preceding discussion define the essential elements for an information-preserving, data-compression system operating under noiseless channel conditions. Another modification which could be an important asset under certain conditions is described in the following.

1. Reduced quantum levels. Observe that the basic operation of the coding system is unaffected by a reduction in the number of quantum levels (i.e., a deletion of the least significant bits). The effect is to move the operating point (in Fig. 8) to the left by an amount equal to the average information contained in the deleted bits. As far as hardware is concerned, this means generating differences from a fewer number of binary digits. The primary benefit realized by the inclusion of this mechanism in the system is an increased compression whenever high-quality TV pictures are not required (e.g., reconnaissance, general planetary topology). For example, a shift from 6-bit PCM to 3-bit PCM will provide *compressed* 3-bit pictures. Another use might be the control of data rates.

2. ***Further development.*** Even under the limited conditions of a noiseless channel, a necessary first step, there still remain many questions and tradeoffs to be considered. For instance, improved dynamic performance curves should be derived and tested on a wide variety of pictures to determine the best block size and thresholds (other codes and extensions can provide additional curves). The possibility of a variable block size is another question. Such tests should include the option of a reduction in quantum levels described above. In addition, the primary hardware components required to implement the algorithm can be identified.

As shown in Section II, the FS portion of the performance curves in Fig. 8 is equivalent to coding a block of TV data with a linear code. This means generating a *Vertical FS* (*VFS*) from the sample matrix in Fig. 1. Further studies should investigate the Code and $\overline{\text{Code}}$ properties of the Modified System using the VFS.

In this report it suffices to mention that a very important problem remaining is the channel coding and decoding of the compressed TV data. The final justification for spacecraft application of the compression algorithm presented here will depend on an adequate solution.

Nomenclature

$\text{Code}(\ell_1, \ell_2, \cdots, \ell_N)$	defines a variable-length code whose first code word has length ℓ_1, whose second has length ℓ_2, etc.
$\overline{\text{Code}}(\ell_1, \ell_2, \cdots, \ell_N)$	equivalent to $\text{Code}(\ell_N, \ell_{N-1}, \cdots, \ell_2, \ell_1)$, and is called a complementary code
FS	fundamental sequence (obtained by performing wiggle operation on input data)
$(\text{FS})^2$	fundamental sequence of the FS
H^*	equals an entropy computed from the distribution of differences for a complete picture (approximates actual entropy if activity in picture is homogeneous)
J	block size
$\hat{H}(n)$	entropy of differences computed for the nth block of J picture elements
$\hat{H}$	average of all the $\hat{H}(n)$; approximates the actual one-dimensional picture entropy
$L(n)$	expected number of bits per pixel required by the coding system to encode the nth block of J picture elements
$\hat{L}$	average of all the $L(n)$

References

1. Anderson, A. T., "A Study of Bandwidth Compression Schemes for Possible Spacecraft Application," in *Supporting Research and Advanced Development*, Space Programs Summary 37-47, Vol. III, pp. 175–183. Jet Propulsion Laboratory, Pasadena, Calif., Oct. 31, 1967.

2. Kleinrock, L., "TV Data Compression by the Slope Threshold Method," in *Supporting Research and Advanced Development*, Space Programs Summary 37-49, Vol. III, pp. 325–328. Jet Propulsion Laboratory, Pasadena, Calif., Feb. 29, 1968.

3. Fadley, J. K., et al., *Video Data Modulation Study: Volume I, Final Report*. NASA Contract NAS 950705, Subcontract NAS 7-100. Electro-Mechanical Research, Inc., Sarasota, Fla., July 31, 1964.

4. Abramson, H., *Information Theory and Coding*. McGraw-Hill Book Co., Inc., New York, 1963.

5. Rice, R. F., *Introduction to the Code Word Shuffle*, General Technical Document 900-138. Jet Propulsion Laboratory, Pasadena, Calif., Apr. 1968.

6. Ash, R. B., *Information Theory*. John Wiley & Sons, Inc., New York, 1965.