A DATA COMPRESSION TECHNIQUE FOR SPACE MONITORING SYSTEMS

Wai-Hung Ng
The Aerospace Corporation
Los Angeles, California

ABSTRACT

For space monitoring systems, it is necessary to compress the transmitted data to minimize the power and bandwidth requirements. Although there are many data compression techniques, here we will only discuss the sample reduction technique because it is the best candidate for cases when the signal contains many redundant samples and the resolution requirement for the reconstructed signal is very high.

In this paper, we introduce a new code to further reduce the compressed signal data derived from the sample reduction technique. The main advantage of this approach is that we can minimize the transmission data rate without requiring a priori knowledge of the signal distribution property. In addition, it can also reduce the buffer size requirement. Examples are given for clarification and discussion.

INTRODUCTION

Data compression is one of the most important elements that space monitoring systems utilize to minimize the required transmission power and bandwidth. From the system's point of view, compressing the transmission data could provide at least four advantages: to speed up the real-time processing, to widen the mission coverage, to improve the resolution of the reconstructed signal, and to enhance the raw data generation rate.

Conceptually, data compression is constructed to reduce the raw data output to the user's required processing rate before transmitting through the channel. Typical raw data sources are two dimensional photographs, vicicon images, voice or speech signals, IR images, etc.

Data compressions could be roughly classified into three basic categories: the source sample encoding, the transform compression, and the redundant sample reduction. The first two categories minimize the data rate by reducing the number of bits required to encode the source samples. But the last one reduces the number of samples before encoding them. In this study, our discussion would concentrate on further improving the compression ratio from the sample reduction technique. Readers who are interested in various data compression techniques should directly refer to the reference list.

The two major redundant sample reduction methods are the prediction method and the interpolation method. In both cases, the algorithm that the data compressor uses to predict the new sample and to decide which sample to transmit must be available to the receiver for data reconstruction. The information needed for the receiver to recognize which samples are transmitted and which samples are deleted is denoted as "time-tag" data. This information has to be transmitted through the channel along with the compressed signal data.

This paper introduces a new coding technique to compress the time-tag data. Regardless of the employed sample reduction method, this technique could provide the following advantages: a) it minimizes the overall transmission data rate, b) its performance is insensitive to the statistical distribution of lengths of intervals between transmitted samples, c) the buffer size requirement could be significantly reduced, and d) the implementation of this approach is relatively simple. Examples are given for illustration and discussion.

REDUNDANT SAMPLE DEDUCTION AND TIME-TAG DATA COMPRESSION

Redundant Sample Reduction Techniques

In communication systems, the signal is sampled and coded with a sampling rate sufficient to provide the maximum rate of signal change. Therefore, many samples which are either introduced by over sampling or able to be reconstructed from the properties of the signal are redundant. Redundancy reduction is theoretically possible since a sample or series of samples can be eliminated from a given input signal and is then reconstructed later by using the samples preceding or succeeding it.

There are many redundant sample reduction methods. From the operational point of view, they could be classified as prediction method and interpolation method. Prediction method eliminates redundant samples through the combination of measuring other samples and comparing them with the present sample. That is, if the difference in the comparison is within a pre-selected value, the present sample is eliminated. This prediction is then propagated to the next sample and the same test is repeated. Whenever the difference exceeds the pre-selected value, the sample will be encoded and transmitted.

The difference between the prediction method and interpolation method is that the later eliminates redundant samples by coding only the end points of signal intervals. Those sample within two end points are eliminated because the deviation of these samples from their interpolated estimates are less than the pre-selected value. In both prediction and interpolation methods, each transmitting sample is encoded into a binary codeword. Typical sample encoding techniques are PCM, DPCM, adaptive PCM, etc.

Zero-order and first-order elimination (i.e., elimination is based on the difference between sample amplitudes or slopes of the samples amplitudes) are the most commonly used prediction/interpolation methods. Occasionally, second or higher-order elimination may be also employed. Among these methods, the zero-order prediction method is the simplest one.

Time-Tag Sequence

Here, we shall describe the basic procedure of using the time-tag data in the sample reduction technique. Let us consider a transmitting sequence consisting of a series of section search containing K samples. Let T be the time-tag sequence where a "1" indicates that a transmitting sample is in this position and a "0" indicates that a compressed sample is in this position. Let S be the binary sample amplitude sequence obtained from the sample amplitude sequence S', in which each sample is encoded into Q binary bits. Without compressing the time-tag sequence, the transmitted sequence, M, consisting of T and S, could be expressed as M = (T) (S). For example, let K = 30 and Q = 6 as shown in Fig. 1, then the M sequence could be expressed as

M = (T)(S) = (101001101010011100011000111011)(000001001100100100100101100001100010
111110111000111000111011111001010000001100001001011110011110).

It is clear that, regardless of how many data samples being compressed, the minimum number of transmitted digits for M is bounded by the number of digits in T, or at least equal to K binary bits. This limitation is a severe drawback for signal S with very high sample compression ratio. For this reason, we introduce a new code here to maximize the data compression ratio. This code would be very useful for compressing signals with high sample compression ratio and random redundant sample distribution as those in space monitoring systems.

Binary Non-consecutive One (BN1) Code

Consider that a typical time-tag sequence contains many blocks of all-zero digits. Let us denote one of these blocks to be X which consists of $x \geq 1$ consecutive zero bits. For example, if X = 0, x = 1; if X = 00, x = 2, and so on. Now, let us encode this X sequence into a BN1 code Y which does not contain consecutive one digit. Let k be the number of binary digits in Y and let $|Y|$ be the decimal value of Y. For example if Y = 01010, it implies that k = 5 and $|Y|$ = 10. Then, the basic encoding rule of this BN1 code is the following.

(i). If X = 0, X is encoded into Y = 0.

(ii). Consider that there are two all-zero sequences X' and X" such that X' is encoded into Y' and X" is encoded into Y". Let x' and x" be the number of zeros in X' and X" such that x' − x" = 1; let |Y| and |Y"| be the decimal values of Y' and Y". Then the following special properties would hold.

a) k' − k" must equal to either 0 or l.
b) If k' − k" = 1, then |Y"| is the maximum decimal value BN1 code with k" binary digits (i.e., Y" is a binary clock sequence as Y" = 101010 ...), and Y' is an all-zero sequence.
c) If k' − k" = 0 and if there is a third BN1 code Y consisting of k' digits and the decimal value is |Y|, then |Y'| > |Y| ≠ |Y"| implies that |Y"| > |Y|.

An example of encoding X into Y is given in Table 1. It shows that (x − k) > 0 for x > 1. Equality holds only when x = 1.

Property of the BN1 Code

The property of this BN1 code could be described by using the combinations of the non-consecutive one digits. Let N_k be the maximum number of different combinations of the k digits long BN1 code, where k > 0, we shall have

$$N_{k+2} = N_{k+}l + N_k \text{ holds for } k > 0.$$

The above statement can be proved by the finite induction theory.

First of all, let us examine three simplest cases: $N_1 = 2$, $N_2 = 3$, and $N_3 = 5$ (i.e., 0 and 1 for N_1; 00, 01, and 10 for N_2; 000, 001, 010, 100, and 101 for N_3). Thus,

a) $N_{k+2} = N_{K+1} + N_k$ is true for k = 1.
b) Assume that when k = n, the equation $N_{n+2} = N_{n+1} + N_n$ is true.
c) Now, we shall prove that when k = n + 1, $N_{n+3} = N_{n+2} + N_{n+l}$ is also true.

Consider that the first digit in the BN1 code is a "0", then the total number of the code sequence is clearly equal to N_{n+2}. However, if the first digit in the BN1 code is a "1", then the second digit can only be a zero and the total number of BN1 code sequence thus equal to N_{n+1}. Therefore, when k = n + 1, the equation $N_{k+2} = N_{k+1} + N_k$ is also true.

By using the same argument, let V_k be the total number of BN1 codes with length less than or equal to k digits long, we shall have

$$V_k = V_{k-1} + N_k = (V_{k-2} + N_{k-1}) + N_k = \sum_{i=1}^{k} N_1$$

The calculation of N_k and V_k is quite easy. For convenience, the values of N_k and V_k for $1 \leq k \leq 20$ are given in Table 2 for ready reference.

$$X = \sum_{i=1}^{k} [(1 + y_i)\ N_{i-1}], \text{ where } N_0 \text{ is defined as } N_0 = 1.$$

The above equations can be easily proved by using the number of different combinations of the BN1 code. For example, if Y = 1010, then x = 2 N_3 + N_2 + 2 N_1 + N_0 = 2x5 + 3 + 2x2 + 1 = 18 (Refer to Table 2). The decimal value x can be converted to the equivalent BN1 code Y with a similar procedure. Since this convertion procedure is straight forward, we would omit in this paper.

REAL-TIME SIGNAL TRANSMISSION

Transmission of Data with BN1 Code

In this section, we shall describe the procedure of transmitting the real-time signal when the BN1 code is being used. Since the BN1 code is of variable length, a “comma” is needed to distinguish and separate it from the binary sample amplitude data. The real-time signal transmission is then conducted as following:

M = ... (BN1 code) (comma) (sample data) (BN1 code)

The comma indicates that the BN1 code has been ended and how many consecutive samples are going to be transmitted. The sample data gives the amplitudes of those transmitted samples. In order to synchronize easily, we would suggest to use a “110" as a comma when there is only one transmitting sample, and a block “2n consecutive one followed by a zero” as a comma when we transmit a block of n consecutive samples. This comma can be easily distinguished from the BN1 code even when the BN1 code ends with a “1” digit. However, if the transmitting samples frequently occur in a solid burst, we should change the comma from “1111...0” to a new comma “0111...0” in which the number of “1” is equal “n+1”. In the following, we shall use a simple example to illustrate this encoding procedure.

Let us refer to the signal data distribution given in Fig. 1. By using the real-time signal transmission format that we just described above, we would have

M = ... (110) (000001) (0) (110) (001100) (1) (11110) (100100100101) ...

For this example, we shall see that the comma is significant not only in separating the BN1 code from the sample amplitude data, but also in stopping the error propagation when errors occur in the receiving sequence.

Comparison of Different Time-Tag Codes

The most commonly used time-tag code is the fixed-length code. In order to maximize the average compression ratio (ACR), we should select the code length $L = \log_2 (ACR)$, and thus L would be longer than 10 bits for (ACR) > 1000. But if we wish to minimize the buffer size requirement, we should select L to be around 3 bits for the lower short time ACR, say (ACR) = 10. To compromise between the ACR and the buffer size, we may select L = 6 bits. A system with a 6-bit time-tag code would limit the maximum sample compression to 63:1, since every 64th consecutive redundant sample must be transmitted as though it were a non-redundant sample. Further discussion on the subject of minimizing the buffer size will be given in the final conclusion.

An example to compare the performance of the fixed-length time-tag code with the BN1 code is given in the following paragraphs.

Assume that there is a block of 10,000 signal samples and each sample is encoded (or quantized) into 6 binary bits. We shall study three situations: no sample compression, sample compression with 6-bit time-tag code, and sample compression with BN1 code for time-tag compression.

a) It is clear that if we transmit the signal without sample compression, the number of digits needed to transmit is equal to 60,000.
b) Assume that an efficient sample compression technique is employed to eliminate 99.9% of the samples. We shall study two extreme cases. First, consider that the transmitted samples are uniformly distributed. We would need 960 bits for time-tag, and 960 bits for samples, or a total of 1920 binary bits. Second, if transmitted samples were all located at the end of the block, we would need 1014 bits for time-tag and 1014 bits for samples, or a total of 2028 binary bits.
c) Consider that a BN1 code is used to reduce the time-tag data in the above case. First, let us assume the 10 transmitted samples are uniformly distributed. In this case, we would need 130 bits for time-tag (i.e., 10 BN1 codes each consists of 13 binary bits), 130 for “commas”, 60 bits for samples, or a total of 220 binary bits. Second, if these

10 samples are located at the end of the block, we would only need 17 bits for one BN1 code, 21 bits for a big comma, 60 bits for samples, or a total of 98 binary bits.

From the above simple example, we can easily see that sample reduction with BN1 code could significantly minimize the required transmission data.

CONCLUSION AND DISCUSSION

In designing the data compression for space monitoring systems, we are most concerned with two critical factors: the long term sample compression ratio and the buffer size requirement.

The sample compression ratio is space monitoring systems is usually very high (> 1000). The system usually utilizes a position code with fixed-length of 12 to 16 binary bits as the time-tag sequence. The main reason for selecting such a long and fixed-length code is to simplify the frame synchronization and to provide a simple one-to-one relationship between each sensor and its corresponding signal position. The system could have a long term minimum data transmission rate providing that: a) one can correctly predict the signal distribution, b) the signal distribution is time-invariant, and c) the system has an infinitive buffer size. But in reality, the signal is random in nature and is usually a time-variant function. Thus a position code with a fixed-length would not be able to satisfy various kinds of unpredictable encountered signals.

We shall compare the typical position code with the BN1 code in two different aspects: the compression ratio and the buffer size requirement.

In designing a system, one should not treat the maximum compression ratio as an absolute value but as a relatively compared value. For example, a compression ratio of 1000 is significantly better than a compression ratio of 125. But the advantage of a system with a long term average compression ratio of 1105 over one with 995 is indeed not too much.

From the coding theory, we know that a fixed length code is only optimum for a signal with a uniformly distributed property. Since the BN1 code is insensitive to the signal distribution, its compression ratio for any random signal in reality is probably very close to the compression ratio of an optimum fixed-length code for signal with uniformly distribution property. Either one could have a slightly higher compression ratio over the other depending upon the actural distribution of the signal.

Now, let us compare the second critical consideration for designing the space monitoring system - the buffer size requirement. As we know, an extensive buffer storage must be used in a data compression system so that the communication channel can cater for the maximum information rate. Let us consider the case that a long burst of dense samples with a short term ACR = 10 must be transmittted through the channel. If a 12-bit long position code is utilized for the time-tag, and if the signal amplitude is encoded into 6 bits per sample, then each of these samples would require 18 bits. However, by using our proposed BN1 coding technique, we would expect that it only needs bits between 9 (i.e., a 3-bit coma “110” and a 6-bit sample amplitude) to 12 (i.e., a 3-bit comma “110”, a 6-bit sample data, and an average of 3 bits long BN1 code for a short term (ACR) = 10) per sample. Based on this rough estimate, we should expect that a 30% to 50% reduction in the buffer size could be achieved by utilizing the BN1 code.

In summary, BN1 code for time-tag data compression is introduced in this paper. The properties of this code would be useful for space monitoring systems. In addition, this code provides a simple synchronization scheme which could allow the length of each codeword to be either fixed or varied.

REFERENCES

1. Davisson, L. D., 1968, “The Theoretical Analysis of Data Compression Systems,” PROC. IEEE.

2. Davisson, L. D., 1972, “Rate Distortion Theory and Applications,” PROC. IEEE.

3. Wintz, P. W., 1972, “Transform Picture Coding,” PROC. IEEE.

4. Stumpe, J. W., 1967, “Redundancy Reduction Techniques and Applications,” Proceeding of NTC.

5. Andrews, C. A., Davies, J. M., and Schwarz, G. R., 1967, “Adaptive Data Compression,” PROC. IEEE.

6. Huffman, D. A., 1952, “A Method for The Construction of Minimum Redundancy Codes,” PROC. IRE.

7. Huang, T. S. and Tretak, O. J. (Editors), 1972, Picture Bandwidth Compression, Gordon and Break, New York.

8. Schreiber, W. F., 1967, “Picture Coding,” PROC. IEEE.

9. Witten, I. H., 1982, Principles of Computer Speech, London, England: Academic.

10. Langdon, G. G., 1983, “A Note on The Ziv-Lempel Model for Comrpressing Individual Sequences,” IEEE Trans. on Information Theory.

11. Jones, C. B., 1981, “An Efficient Coding System for Long Source Sequences,” IEEE Trans. on Information Theory.

12. Raviv, J., 1967, “Decision Making in Markov Chains Applied to The Problem of Pattern Recognition,” IEEE Trans. on Information Theory.

TABLE 1   ENCODING OF X INTO THE BN1 CODE Y

x	y	k	(x - k)
1	0	1	0
2	1	1	1
3	00	2	1
4	01	2	2
5	10	2	3
6	000	3	3
7	001	3	4
8	010	3	5
9	100	3	6
10	101	3	7
11	0000	4	7
12	0001	4	8
13	0010	4	9
14	0100	4	10
15	0101	4	11
16	1000	4	12
17	1001	4	13
18	1010	4	14
19	00000	5	14
20	00001	5	15

TABLE 2 MAXIMUM NUMBER OF BN1 CODE FOR k = 1 – 20

k	$N_k = N_{k-1} N_{k-2}$	$V_k = \sum_{i=1}^{k} N_i$
====	=============	=============
1	2	2
2	3	5
3	5	10
4	8	18
5	13	31
6	21	52
7	34	86
8	55	141
9	89	230
10	144	374
11	233	607
12	377	984
13	610	1594
14	987	2581
15	1597	4178
16	2584	6762
17	4181	10943
18	6765	17708
19	10946	28654
20	17711	46365

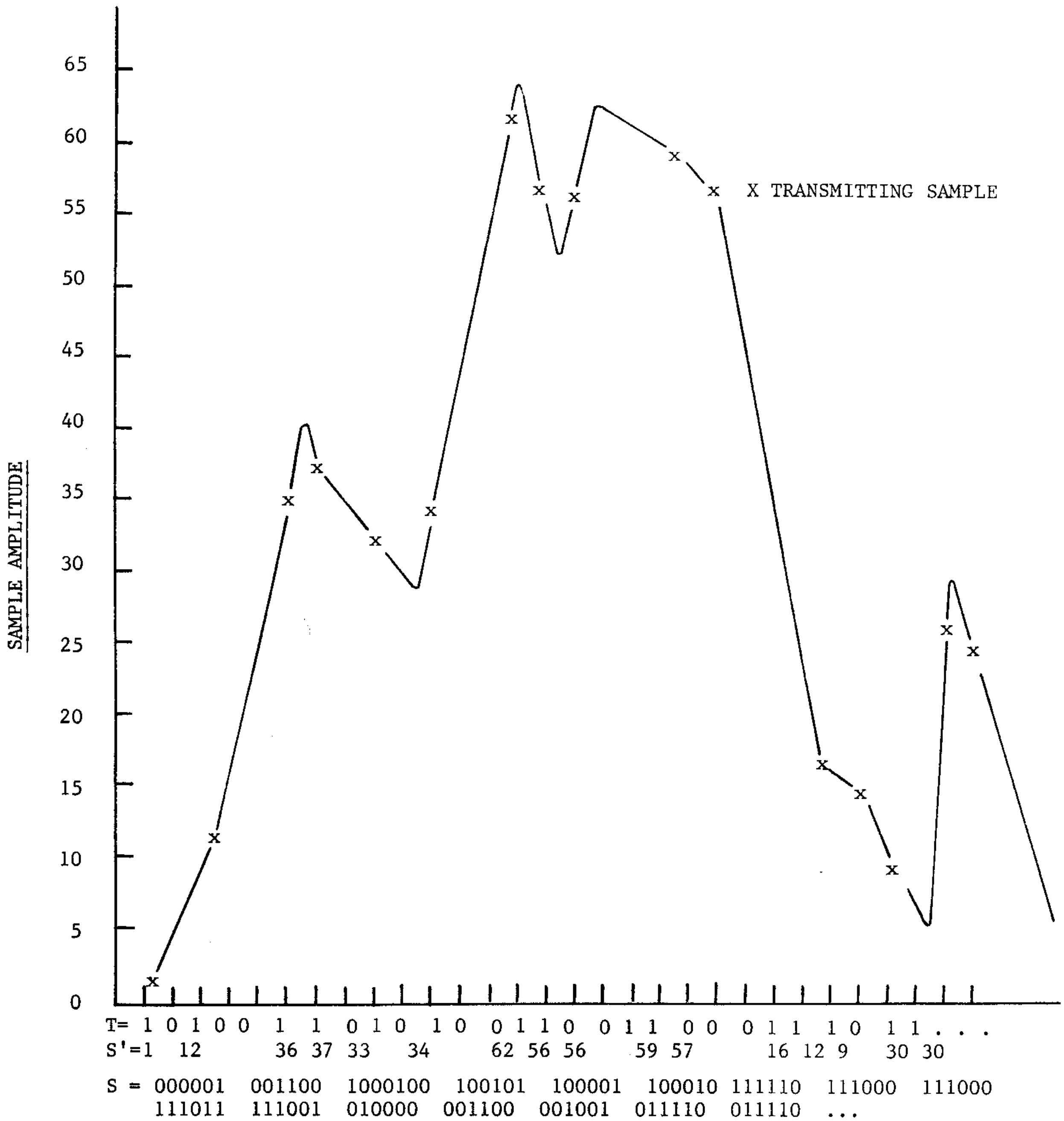

Figure 1. Data Sampling and Data Compression