54-82

3,3877

208065

1997

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

DATA COMPRESSION TECHNIQUES
FOR
ADVANCED SPACE TRANSPORTATION SYSTEMS

Prepared by: William G. Bradley, Ph.D.

Academic Rank: Professor of Electrical Engineering

Institution and Department: Western New England College
                                     Department of Electrical Engineering

NASA/MSFC

    Office: Astrionics Laboratory
    Division: Computers and Data Systems

MSFC Colleague: Leann Thomas

IV
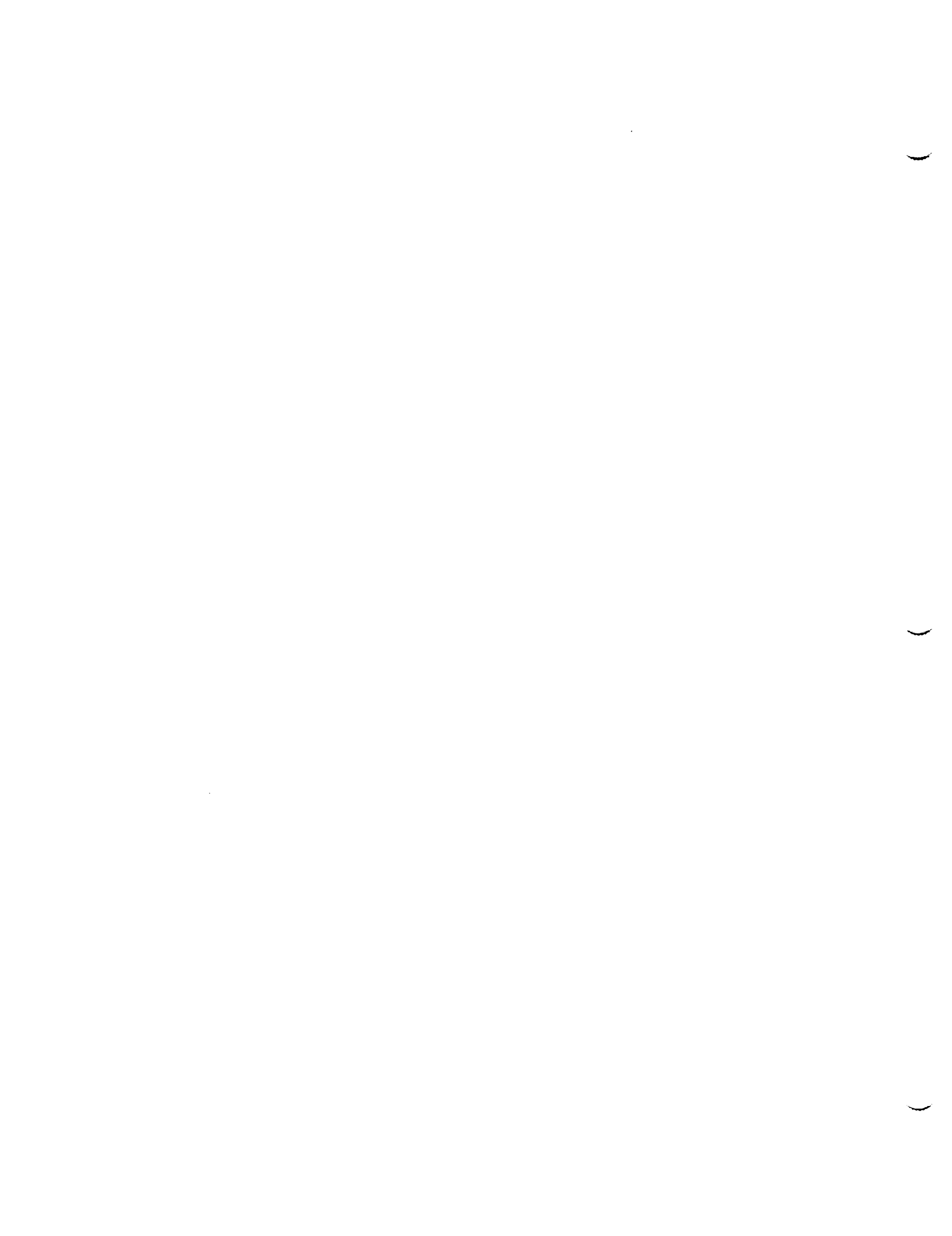
Introduction

Advanced space transportation systems, including vehicle state of health systems, will produce large amounts of data which must be stored on board the vehicle and or transmitted to the ground and stored. The cost of storage or transmission of the data could be reduced if the number of bits required to represent the data is reduced by the use of data compression techniques. Most of the work done in this study was rather generic and could apply to many data compression systems, but the first application area to be considered was launch vehicle state of health telemetry systems.

A very large amount of information on data compression is available in journals, in books and on the Internet. The book Introduction to Data Compression (Sayood 1996) was used as an introduction to the broad field.

Both lossless and lossy compression techniques were considered in this study. Lossless data compression guarantees that no information is lost. This means that the original signal can be reconstructed from the compressed signal with no distortion. Lossy compression may introduce some distortion when the signal is reconstructed from the compressed version, but has the potential of much higher compression ratios (The ratio of the number of bits before compression to the number of bits after compression) than lossless compression when a small amount of distortion can be tolerated. "Lossless compression is generally used with "discrete" data such as text, computer generated data, and some kinds of image and video information" (Sayood 1996, p. 3) Lossy compression must be used very carefully, if at all, on signals where post processing techniques may be used to enhance the data. The post processing may "enhance" small differences between the original signal and the reconstructed signal (Sayood 1996, pp. 4-5). However, there are many applications where small differences between the original and reconstructed signals are acceptable. For example, in video and image processing small distortions are acceptable if they are not seen by the human eye, and lossy compression is usually used. Sensor data often starts as an analog signal, and the digitized signal is an approximation of the analog signal. Also, in many cases, the signal contains a significant amount of noise. The lossy reconstruction of a noisy signal may be better than the original if noise is removed.

Lossless Data Compression

Lossless Data compression techniques are subdivided into dictionary codes and entropy codes. There are several types of **dictionary codes**, but they all work by looking for repeating patterns or words in the data and assigning them special codes. Some applications that use dictionary codes include; 1) ZIP and PKZIP that are used to compress PC files and were tested on telemetry data as part of this study, 2) UNIX Compress algorithm, 3) GIF used to compress graphics images, and 4) V.42 used in modems. **Entropy codes** work by assigning short code words to the most common letters (or bit patterns in binary data), and longer code words to less common letters. The most common entropy code is the Huffman code. It gives optimum results if the statistics of the data are known is advance. Unfortunately the statistics of scientific data are seldom known in advance. There are adaptive Huffman codes that adapt to the statistics of the data, but the code trees get very cumbersome if the alphabet is large. Arithmetic coding is

another type of entropy coding that has become increasingly popular recently (Sayood 1996, Ch. 4). The entropy code that was given the most attention is this study is the Rice algorithm.

Rice Algorithm

The Rice algorithm was initially developed by Robert Rice at JPL (Rice et el. 1971, 1991). A research group at Goddard Space Flight Center, including Pen-Shu Yeh, refined the algorithm, and called it "Universal Source Encoding for Space" (Yeh and Miller Dec. 1993). It was shown to produce optimum code for the special case of Laplacian distributed data (Yeh et al. Oct. 1993). The Rice Algorithm has been adopted as a standard for space data systems (Consultative 1995). It is highly adaptive and suitable for real time high speed applications. The algorithm has been implemented in a chip called USES (Universal Source Encoding for Science Data) that is available from the Microelectronics Research Center at the University of New Mexico (MRC 1997). This chip has been used on several space missions.

Software called 'szip' that simulates the USES chip can be downloaded from the Microelectronics Research Center web site (MRC 1997). It was downloaded along with some sample test data. When szip was used to compress the sample test data, the results were identical to those given on the web site. The only discrepancy was that it was observed that the seismic data had been zero padded with about ten thousand zeros, and when the zeros were removed the compression ratio dropped from 1.97:1 to 1.68:1.

The Rice algorithm consists of a predictor, a mapper, and an entropy coder. Any prediction algorithm could be used with the Rice Algorithm, but the USES chip and the szip software use a simple nearest neighbor predictor. The mapper maps the output of the predictor into the standard format required by the entropy coder. The standard format requires that the data must be all positive integers with zero the most probable integer and the probability of occurrence of other integers must decrease with their magnitude. If adjacent samples of the data are correlated, use of the predictor improves the compression ratio. However, if adjacent samples are uncorrelated, use of the predictor decreases the compression ratio, and better results are obtained by bypassing the predictor. If the predictor is by passed, the user must provide his own algorithm to map the data into the standard format required by the entropy coder. The data is divided into blocks and each block is coded with up to 8 different coding options. The option that yields the highest compression ratio is chosen for that block of data. The default block size is 16, but other block sizes may yield higher compression ratios for some types of data.

Lossy Compression with Wavelets

The wavelet transform is an invertable transform (Rioul and Vetterli 1991, Strang and Nguyen 1996) However, losses may occur due to computational accuracy and quantization of the wavelet coefficients. Wavelet compression is possible because of the relative scarceness of the wavelet domain representation of the signal. This allows compression of the wavelet coefficients because the original signal can be approximated by a small number of approximation coefficients at an appropriate level and some of the detail coefficients with the rest of the detail coefficients set to zero (Misti et al. 1996 p. 6-86).

The wavelet computations in this sturdy used the MATLAB Wavelet Tool Box and the 1-D Graphical Tool. (Misti et al. 1996). Use of the Graphical Tool made it possible to try many different wavelets at different levels in a short time. However, implementation of wavelet compression will require work with the details that MATLAB performs automatically.

Engine Data Test Results

The first test of the compression algorithms was on state of health telemetry data from the DC-XA that was recorded by Lisa Blue. Of the many variables recorded, a rocket engine vibration trace was chosen for initial testing. This trace was chosen because its sampling rate of 12,000 samples per second was the highest of the available data, and it appeared to be 'noise like' data and was thought to be the hardest to compress. Unfortunately, after most of the testing was done, it was determined that the trace contained only every third sample of the original data. This means that it was the engine vibration data used was effectively sampled at 4000 samples per second. It is believed that the compression ratios would have been somewhat higher if all of the samples had been available. A subset of 32,768 samples from the test was used to obtain the results shown in table 1. When using the Rice Algorithm with the predictor, 128 was added to all samples to make the data positive integers as required by the szip software. When the Rice Algorithm was used without the predictor, a program was needed to map the data into the standard format required by the entropy coder. A MATLAB m-file was written and used as the mapper. When using the Rice Algorithm, the block size is a variable. The default block size is 16, but for some cases, block sizes of 8 and 4 were tried. When using wavelets, the wavelet coefficients were quantized to 8 bits and coded with the Rice Algorithm and PKZIP. The coefficients were then thresholded at two different levels and then coded with the Rice Algorithm and PKZIP. Space limitations do not permit including the original and reconstructed waveforms, but they are in a Power Point presentation that is available from the author at wbradley@wnec.edu. Wavelet Packet Analysis was also tried on this data. However, the considerable extra complexity of the Wavelet Packet Analysis does not appear to be justified by the very small increases in the compression ratio that were obtained.

	Original Data	Wavelet Coef	Thresholded Coef 90.1% Retain En 80.1% Zeros	Thresholded Coef 85.1% Retain En 85.1% Zeros
Rice With Predictor	1.21	1.24	-	2.91
Rice (WO pred) Block Size 16	1.22	1.40	2.88	3.24
Rice (WO pred) Block Size 8	-	1.37	2.89	3.38
Rice (WO pred) Block Size 4	-	-	2.93	3.48
PKZIP	1.22	1.34	3.92	5.02

Table 1 Compression Ratios for Engine Data

The compression ratios in table 1 are relatively modest. However, this is believed to be a worst case example, and the compression ratios should be higher for most other data. The fact that PKZIP did significantly better that the Rice Algorithm when coding the wavelet coefficients indicates that there probably are better lossless algorithms than the Rice Algorithm for coding the wavelet coefficients.

Other Data from DC-XA

In addition to the engine vibration data, one temperature, one pressure and one strain trace from the DC-XA test were obtained. Not much time was available for analyzing this data, but a few results were obtained. The 8 bit pressure data was compressed with the Rice Algorithm resulting in a compression ratio of 1.26:1 and with PKZIP resulting in a compression ratio of 1.31:1. When a 1024 byte subset of the pressure data was analyzed with the MATLAB 1-D Compression Tool, 97.52% of the coefficients were zeroed. However, not enough is known about the actual pressure to tell whether the high frequency fluctuations that were removed by the wavelet compression are noise or actual pressure fluctuations.

AR&C Data Test Results

Data from an AR&C (Automatic Rendezvous and Capture) system simulation and some unfiltered sensor data was obtained from Richard Dabney. The data was state of health information from an automatic docking system. This was relatively slowly varying data that was sampled at 20 samples per second and stored as floating point numbers. The Rice algorithm was not used on this data because it would require that the data be scaled and converted to integer form. However, the MATLAB Wavelet Tool Box was used to analyze the data.

Three separate signals were analyzed. Each signal contained 7,989 data points. Several different wavelets and different levels of decomposition were tried on each signal. The exact wavelet used and level of decomposition was not very critical, but it was found that the Daubechies 3 or 5 wavelets at levels 7 or 9 worked as well as any that were tried. The first two traces were from a simulation and contained little noise.

- The first signal was a relative distance measurement. When compressed using the Daubechies 5 wavelet at level 7, 99.04% of the coefficients were zeroed while retaining 99.06% of the signal energy. The second was a differential distance measurement. When compressed with the Daubechies 5 wavelet at level 9, 99.5% of the coefficients were zeroed while retaining 99.49% of the signal energy. In both cases the reconstructed signal appeared to be an excellent approximation to the original signal.
- The third signal to be analyzed was very interesting because it was unfiltered altitude sensor data that contained a significant amount of noise. When compressed with the Daubechies 3 wavelet at level 7, 98.87% of the coefficients were zeroed while retaining 98.7% of the signal energy. **When the signal was reconstructed, almost all of the noise was removed.**

It is believed that a code other than the Rice algorithm should be used to code the wavelet coefficients when most of the coefficients have been zeroed. Perhaps an identifier tag could be added to the coefficients that are retained, and the zeros not transmitted.

Conclusions

1) Lossless compression with the Rice Algorithm gives modest compression ratios.
   - Compression ratios of only about 1.2:1 to1.3:1 were obtained with the Rice Algorithm on the data tested. Somewhat higher compression ratios would be possible if the data had more correlation between samples.
   - It is not clear whether or not the relatively small amount of compression is enough to justify the extra complexity that would be required to handle the varying word length, and possible error propagation due to the lossless coding.
   - PKZIP did as well or a little better than the Rice Algorithm.
2) Wavelet compression should be considered for higher compression ratio.
   - Compression Ratios of 3:1 to 5:1 were obtained on the engine vibration data with 85% of the coefficients zeroed.
   - 98.9 to 99.5% of the coefficients were zeroed with the AR&C (autodock) data.
   - Most of the noise was removed from a noisy AR&C signal.
   - Two dimensional wavelet analysis could be tried as has been used successfully with seismic data (Aware 1997)
   - Much additional study of wavelet compression is needed including more testing and better methods of coding wavelet coefficients.

References

Aware Inc. (1997), SeisPact Wavelet based Seismic Data Compression, http://www.aware.com/

Consultative Committee for Space Data Systems (1995), Universal Source Coding for Data Compression, CCSDS 121.0-R-1 RED BOOK, Nov. 1995.

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

Rice, R. F., Yeh, P. S. and Miller, W. (1991), Algorithms for a Very High Speed Universal Noiseless Coding Module, JPL Publication 91-1, 1991.

Rioul, O. and Vetterli, M. (1991), "Wavelets and Signal Processing", IEEE Signal Processing Magazine, Oct. 1991, pp. 14-38.

Sayood, K. (1996), Introduction to Data Compression, Morgan Kaufmann Publishers, Inc., San Francisco CA

Strang, G. and Nguyen, T. (1996), Wavelets and Filter Banks, Wellesley-Cambridge Press, Wellesley MA.

Microelectronics Research Center (MRC, 1997), Web Site 'http://www.mrc.unm.edu'

Misti, M., Misti, Y., Oppenheim, G. and Poggi, m. (1996), Wavelet Tool Box for use with MATLAB, The Math Works, Inc., Natic MA

Yeh, P. S., Rice, R. F. and Miller, W. H.(Oct. 1993), On the Optimality of Code Options for a Universal Noiseless Coder, revised version, JPL Publication 91-2, 1991, a shortened version in Proceedings of the AIAA computing in Aerospace 9 Conference, San Diego, CA, Oct. 19-21, 1993.

Yeh, P. S. and Miller, W. H. (Dec. 1993), Application Guide of Universal Source Encoding for Space, NASA Technical Paper 3441, Dec. 1993.