IMPLEMENTATION OF CCSDS LOSSLESS DATA COMPRESSION FOR SPACE AND DATA ARCHIVE APPLICATIONS

Pen-Shu Yeh[*]
NASA/Goddard Space Flight Center

ABSTRACT

Goddard Space Flight Center actively participated in the mid 90's in an effort to standardize a lossless data compression algorithm for space applications. As the standard effort progressed, implementation in Application Specific Integrated Circuit (ASIC) was initiated for high throughput applications. Eventually, a radiation hardened circuit was fabricated to function at over 20 Msamples/sec.

Implementation of new technologies into space missions has always met resistance. The mentality of "If it works, why needs change?" prevails in aerospace community. The notion of "not-invented-here", or otherwise known as NIH disease further hampers progress. The first real mission application at GSFC of the lossless standard was for a small explorer, the Sub-millimeter Wave Astronomy Satellite (SWAS-1999) that needed to overcome insufficient onboard storage capacity. Subsequent mid-class explores for space science missions, Imager for Magnetopause-to-Aurora Global Exploration (IMAGE-00), Microwave Anisotropic Probe (MAP-01), followed. These implementations are all software based.

Migration of new standards and technologies into larger class space missions is shown to be even more difficult. Until today, none of the launched NASA Earth observing missions has lossless compression onboard. The delay can be attributed to the long development time of larger class missions, the unwillingness to take risk and the lack of technology validation opportunity.

In this presentation, the history of the technology development which preceded and later supported the standard activity will be reviewed along with more recent technology evolution. Examples of known mission implementations will be given which includes both space applications and ground archive use.

1. THE CCSDS LOSSLESS DATA COMPRESSION RECOMMENDATION

1.1 Development history of the recommendation

In 1992, GSFC proposed to CCSDS panel 1 to start a new work item in lossless data compression for space missions. A presentation was made to the panel on proposing the Rice algorithm [1] as a candidate due to its excellent performance and low complexity for implementation. The proposal did not gain much support probably because bandwidth and onboard storage were not major issues facing the international aerospace community then.

Meanwhile GSFC was responsible for managing the main instrument, the MODerate Resolution Imaging Spectroradiometer (MODIS) for the first flagship in the series of the Earth Observing Systems (EOS) missions. MODIS has 36 spectral channels. Several scientists at GSFC had raised the concern of not being able to download all mission data within the allocated bandwidth. This prompted us to re-visit the issue of establishing a compression standard for space use.

---

[*] pen-shu.yeh@gsfc.nasa.gov; phone 1 301 286-4477; fax 1 301-286-0220; Goddard Space Flight Center, Code 564, Greenbelt, MD, 20771 USA

In the fall of 1994, GSFC re-proposed to panel 1. The work item was accepted by all member agencies and the proposed algorithm was quickly adopted. In 1997, CCSDS published a Blue Book on a recommendation for lossless data compression [2]. An accompanying Green Book was also released to provide guidelines for system designers [3].

In proposing the lossless compression work item, requirements were first established which include:

a. The algorithm has to adapt to the changes in data statistics to maximize compression performance.
b. The algorithm must be compatible with a real time implementation while minimizing memory and power budgets.
c. The algorithm must interface with a packet data system such that each packet can be independently decoded without requiring information from other packets.

Several available lossless algorithms were compared on test data. The results strongly supported the adoption of the Rice algorithm as a recommendation.

1.2 Algorithm description

The recommendation is based on the Rice algorithm [1] with added low entropy options. The algorithm exploits a set of variable-length codes to achieve compression. Each code is nearly optimal for a particular geometrically distributed source. By using several different codes and transmitting the code identifier, the algorithm can adapt to many sources from low entropy (more compressible) to high entropy (less compressible). Because blocks of source samples are encoded independently, side information does not need to be carried across packet boundaries if de-correlation of source samples are only executed among these blocks, then the performance of the algorithm is independent of packet size.

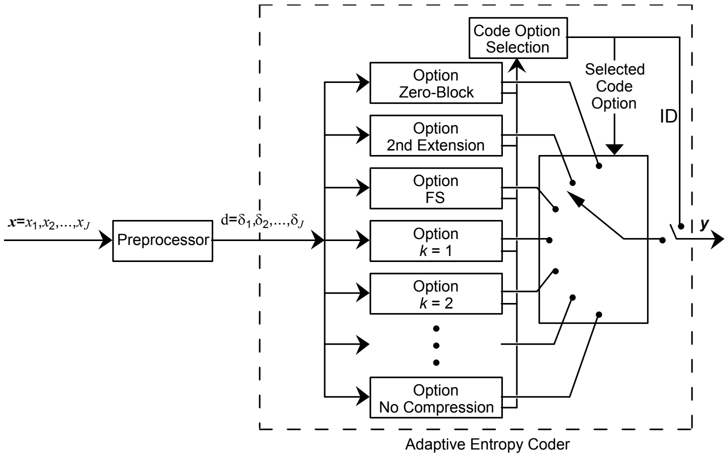


Fig. 1-1: The encoder architecture

A block diagram of the algorithm is shown in Figure 1-1. It consists of a preprocessor to de-correlate a block of *J*-sample data and subsequently map them into symbols suitable for the entropy coding stage. The entropy coding module is a collection of variable-length codes operating in parallel on blocks of *J* preprocessed samples. The coding option achieving the highest compression is selected for transmission, along with an ID bit pattern used to identify the option to the decoder. Because a new compression option can be selected for each block, the algorithm can adapt to changing source statistics. Although the CCSDS Recommendation specifies that the parameter *J* be either 8 or 16 samples per block, the preferred value is 16. The value of 16 samples per block is the result of experiments performed on several classes of science data, both imaging and non-imaging. These studies monitored the achievable compression ratio as a function of the parameter *J*, which was set to 8, 16, 32, and 64 samples/block for the various classes of science data. Values of *J* less than 16 result in a higher percentage of overhead, which yields a lower compression ratio, whereas values of *J* higher than 16 yield low overhead but have less adaptability to variations in source data statistics.

1.2.1 The fundamental sequence encoding

The most basic code construct in the algorithm is the *Fundamental Sequence* code which is also known as the “comma code”. In this code, the codeword contains a string of “0” digits equal to the decimal value of the symbol to be coded. It then uses digit “1” to signal the end of a current codeword. This simple procedure allows FS codewords to be decoded without the use of lookup tables.

Table 1-1 illustrates the FS codewords for preprocessed sample values with *n*-bit dynamic range.

Preprocessed Sample Values, δ_i	FS Codeword
0	1
1	01
2	001
.	.
.	.
.	.
2^n-1	$\underbrace{0000\ldots00001}_{(2^n-1 \text{ zeros})}$

Table 1-1: Fundamental sequence codewords as a function of the preprocessed samples

1.2.2 The split-sample option

Most of the options in the entropy coder are called ‘split-sample options’. The k^{th} split-sample option takes a block of *J* preprocessed data samples, splits off the *k* least significant bits (LSB) from each sample and encodes the remaining higher order bits with a simple FS codeword before appending the split bits to the encoded FS data stream. This is illustrated in Table 1-2 for the case of *k* split bit being 0 (no split-bit), 1 or 2 on a sequence of 4-bit samples. From Table 1-2 either $k = 1$ or 2 will achieve data reduction from the original 32 bits to 29 bits. As a convention, when a tie exists, the option with smaller *k* value is chosen. In this case, $k = 1$ will be selected. When a block of *J* samples are coded with one split-sample option, the *k* split bits from each sample are concatenated.

Each split-sample option in the algorithm is designed to produce compressed data with an increment in the code word length of about 1 bit/sample (approximately $k + 1.5$ to $k + 2.5$ bits/sample) [3]; the code option yielding the fewest encoded bits will be chosen for each block by the option-select logic. This option selection process assures that the block will be coded with the best available code option on the same block of data, but this does not necessarily imply that the source entropy lies in that range. The actual source entropy value could be lower; the source statistics and the effectiveness of the preprocessing stage determine how closely entropy can be approached.

Sample Values	4-bit Binary Representation	FS Code, $k = 0$	$k = 1$ 1 LSB + FS Code	$k = 2$ 2 LSB + FS Code
8	1000	000000001	0 00001	00 001
7	0111	00000001	1 0001	11 01
1	0001	01	1 1	01 1
4	0100	00001	0 001	00 01
2	0010	001	0 01	10 1
5	0101	000001	1 001	01 01
0	0000	1	0 1	00 1
3	0011	0001	1 01	11 1
Total Bits	32	38	29	29

Table 1-2: Examples of split-sample options using fundamental sequence codes

2. LOSSLESS COMPRESSION TECHNOLOGY DEVELOPMENT AT GSFC

The lossless compression technology development at GSFC started with analytical work on the algorithm and proceeded further into hardware and software implementation.

2.1 Algorithm evolution

In 1989, a study was initiated to devise an optimal solution for onboard lossless data compression. The Rice algorithm was thoroughly analyzed based on its early form published in 1971 [4] and its variations in the following 20 years [5][6]. A mathematical analysis was established that proves each of the split-sample options in the Rice algorithm is essentially a Huffman code for a data source with Laplacian probability distribution function (PDF) at a given entropy range [7]. With this proof, we defined a Rice coding scheme to contain only the split-sample options for ease of implementation. A low-entropy extension was added later to provide performance enhancement at entropy range below 1 bits-per-pixel (bpp). The performance of the scheme was also provided for Gaussian PDF and Poisson PDF [7]. Basically, the algorithm works well when the data source is uni-modal with decreasing PDF. For most of spacecraft instrument data, the data source can fit well in this model with an appropriate prediction scheme.

With the defined Rice algorithm, simulations on test data became much easier and performance comparison with other known techniques became possible.

2.2 Implementation

2.2.1 Hardware development

Once the algorithm performance was established, effort for prototyping in hardware commenced in 1990. An engineering prototype chipset based only on the split-sample code options was designed and fabricated in 1.0u CMOS process. These chip set obtained over 50 Msamples/sec processing speed at less than 1.0 watts power consumption [8]. However, this prototype encoder was designed for proof-of-concept and required an outside packetizer to concatenate coded data blocks into a longer packet of bit string.

The prototype chip set was integrated into a communication test, which successfully demonstrated the transfer of losslessly compressed images in an end-to-end system in 1994. In this test, the compressed data was formatted into variable length CCSDS packets in the Advanced Orbiting System Test-bed (AOST). The coded virtual channel data units were transferred to the RF Simulations Operations Center (RFSOC) via fiber optic link, where data was then transmitted through the geo-synchronous Tracking and Data Relay Satellite System (TDRSS) and back to the AOST via the White Sands Complex (WSC) [9].

While the engineering work was proceeding in the AOST, a flight version of the lossless compressor was designed which circumvented the need for an off-chip packetizer. In addition, the low-entropy code option was added. This new chip, the Universal Source Encoder for Space (USES), was targeted for space flight and was fabricated in a proven radiation hardened circuit in 1994. USES offers over 20 Msamples/sec processing speed with 0.1 watts/Msample/sec power consumption and input data quantization up to 15 bits/sample. A matching decoder chip was later completed in 1997.

It is recognized that a new circuit able to take data of quantization at 16-bit or larger will eventually be needed for flight missions. Such an implementation without the low entropy option in radiation hardened field programmable gate array (FPGA) is underway. We are currently actively seeking opportunity to develop a new ASIC that will take larger than 15-bit input and also compress data of floating point format.

Lately a new implementation of the lossless compression USES was executed to validate our latest space electronics development in ultra low power RT technique. A 3.3V RT part now can process up to 80 Msamples/sec at 15 mwatts/Msample/sec, and the 0.5V RT part consumes less than 0.5 mwatts/Msamples/sec processing speed. We expect after full radiation characterization, these low power chips will benefit most future missions.

2.2.2 Software development

As part of the hardware development process, software module was written to crosscheck the correctness of hardware design. This was written in C language and was improved over the years that it is currently viable for integrating into spacecraft data processing module for missions that do not require high-throughput rate.

3. MISSION INFUSION AND APPLICATIONS

With a published CCSDS recommendation in lossless compression, it still does not guarantee acceptance by space missions. We have learned from our experience that to infuse a standard into missions, we need to have three essential elements:

a. Technology: either software or hardware has to be developed and available for integrating into space flight sub-systems.
b. Support services: simulation support and question-and-answer service has to be available to missions.
c. Flight validation: this provides heritage for missions.

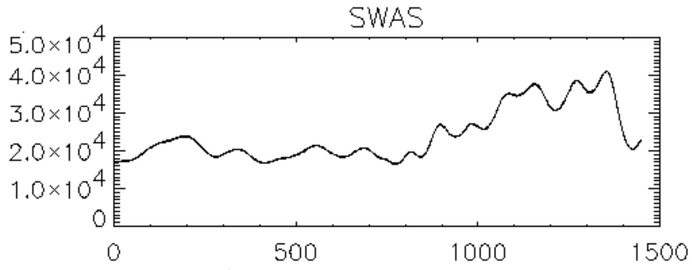
The infusion of a recommendation can also benefit greatly by publicizing the developed technology in technical conferences. The first usage of the USES chip on a spacecraft (1997) is a result of this activity, whereas the first application of the compression software resulted from providing simulation support to an in-house project at GSFC (1999).

3.1 Infusion examples

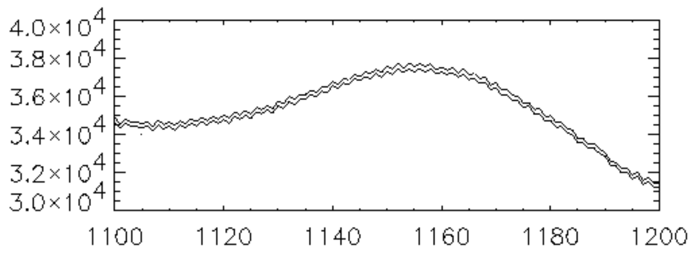
In providing simulation support to flight projects, one cannot blindly apply compression to test data. Instead, more optimal compression configuration is often devised by taking into considerations:

a. types of instrument and data acquizition constraint,
b. packet data architecture,
c. degree of tolerance to error propagation.

A good example is illustrated in compressing data from the acousto-optical spectrometer on the Sub-millimeter Wave Astronomy Satellite (SWAS, launched 1999). The spectrometer waveforms are shown in Figure 3-1. The upper graph contains two traces. Each trace contains data from two different analog-to-digital converter (ADC) outputing data in alternate fashion, the expanded view in the lower graph shows the saw-tooth shaped curves resulting from the different electronic characteristics of the two ADCs. The ADC output 16-bit data. If we apply a simple nearest-neighbor predictor, or a predictor using the previous trace, the compression ratio is in the range of 1.6, or equivalently 10 bpp. However, knowing that the ADCs have different offset which may change with temperature variation, a "double-difference" predictor scheme is applied and the compression ratio increases to 2.3 (6.9 bpp).



(a). A typical spectral trace



(b). Two traces expanded

Figure 3-1 Acousto-optical spectrometer data traces

In a different application, the compression hardware would be tested on data from the hyper spectral imager (HSI) on the Lewis mission[1] which was designed to flight validate technologies. The data is spatial-spectral in nature since the imager has 128 channels in the visible-near-infra-red spectral region and 256 channels in short-wave-infra-red region. A spatial-spectral domain data is shown in Figure 3.2. If compression is applied along the spatial domain, a compression ratio of 2.6 (4.6 bpp) is obtained for the

---

[1] Lewis mission was lost during orbit insertion.

simulated 12-bit data. Our study showed that with the "double-difference" predictor scheme applied in both spectral and spatial domains, the compression ratio increases to 3.4.

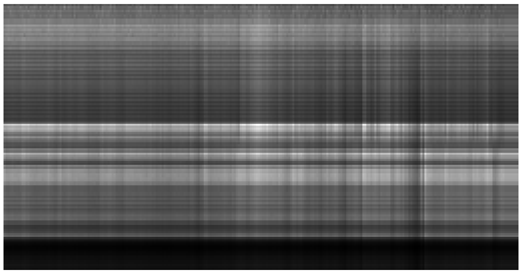


Figure 3-2. Simulated hyper spectral imager data

3. 2 Application summary

Over the last few years, we have performed significant amount of simulation support to projects that are either in planning stage or in execution phase. Table 3-1 provides a summary of mission applications that have used the CCSDS lossless data compression.

Mission	Launch	Lead Agency	Implementation
SERTS-97 (Sounding Rocket)	11/97	NASA/GSFC	HW
COBRA	/97	DOE/USA	HW
LEWIS/SSTI	0/97	NASA	HW
CASSINI CDA	10/97	NASA/JPL	SW upload after launch
SWAS/SMEX-3	01/99	NASA/GSFC	SW
KOMPSAT-1	/99	KARI	HW
IMAGE/MIDEX-1	02/00	NASA/JPL	SW
THEMIS/Mars Odyssey	04/01	NASA/JPL	HW
MAP/MIDEX-2	07/01	NASA/GSFC	SW
EOS-CHEM/AURA	/02	NASA/GSFC	HW
VCL/ESSP-01		NASA/GSFC	HW
ROSETTA	01/03	ESA	HW
SBIRS	Multiple	DOD/USA	HW
INTEGRAL SPI	/03	CNES	SW
MESSENGER MLA	/03	NASA/GSFC	SW
GIFTS/EO-3	/04	NASA/LaRC	HW
PICARD	/05	CNES	SW/DSP
NPP	/	NOAA/NASA	HW
NGST	/	NASA/GSFC	HW
ESDIS/HDF --archive	/03	NASA/GSFC	SW

Table 3-1 CCSDS lossless data compression Applications

Besides the obvious benefits to space missions, lately we have embarked on an effort to infuse the compression technique into ground data distribution and archive facilities [11]. The completion of the project will reduce not only the archive volume, but also the network connection time needed for distributing science data product over the internet.

From Table 3-1, it is seen that most of the applications before 2002 are space science missions. Migration of new standards and technologies into larger class Earth science missions has not been an easy task. Until today, none of the launched NASA Earth observing missions has lossless compression onboard. The delay can be attributed to the long development time of larger class missions, the unwillingness to take risk and the lack of technology validation opportunity. However, with heritage provided by space science missions and an internationally recognized CCSDS standard, most of future Earth observing programs have mandated the use of compression in early planning stage.

4. CONCLUSION

The lossless data compression algorithm recommended by CCSDS has benefited many space missions by either reducing bandwidth, onboard storage requirement, or by increasing science data return. The percentage data reduction may not be the best achievable considering all other available techniques; however, its simplicity and adaptivity does allow high-speed space implementation and applicability to not only image data, but data emanating from the vast diversity of instruments including interferometer, altimeter and spectrometer.

We have learned from past experience that a standard alone will not guarantee its acceptance for space missions. Its application has to be backed by viable technology development, credible flight validation and expertise support service. In fact, the development of a new standard preferably should start with research and technology effort that will point to a practical implementation scheme to avoid unnecessary revisions of the standard later.

ACKNOWLEDGEMENT

The author acknowledges her late colleague Warner H. Miller at NASA for the initiation of the compression panel work and his guidance throughout the course of technology development. The author also acknowledges the effort from the following colleagues: Jack Venbrux, Lowell Miles at the University of Idaho for their major contributions to the ASIC and software development; Joe Miko, Wai Fong at GSFC for leading the AOST test in addition to implementing for the SERTS and LEWIS missions and Wei Serafina at GS&T for providing support for ground archive application.

References

1. R. F. Rice, P-S Yeh and W. Miller, "Algorithms for a very high speed universal noiseless coding module", JPL Publication 91-1, Jet Propulsion Laboratory, Pasadena, CA, Feb., 1991.
2. *Lossless Data Compression*, Recommendation for Space Data Systems Standards, CCSDS 121.0-B-1, Issue 1, May 1997.
3. *Lossless Data Compression*, Green Book, CCSDS 120.0-G-1, May 1997.
4. R. F. Rice and J. R. Plaunt, "Adaptive variable length coding for efficient compression of spacecraft television data," *IEEE Trans. on Communication Technology*, V. COM-19, Part 1, Dec. 1971, pp. 889-897.
5. R. F. Rice, "Some practical universal noiseless coding techniques," JPL Publication 79-22, Jet Propulsion Laboratory, Pasadena, CA, March, 1979.
6. R. F. Rice and J. Lee, " Some practical universal noiseless coding techniques," Part II, JPL publication 83-17, Jet Propulsion Laboratory, Pasadena, CA, March, 1983.

7. P.-S Yeh, R. F. Rice and W. Miller , "On the optimality of a universal noiseless coder," *Proc. of AIAA Computing in Aerospace 9 Conf.* San Diego, CA., Oct, 1993. Also as JPL Publication 91-2, Feb. 1991.
8. J. Venbrux, P.-S. Yeh and M. N. Liu, "A VLSI chipo set for high-speed lossless data compression," *IEEE. Trans. on Circuits and Systems for Video technology,* V. 2, No. 4, Dec, 1992, pp. 381-391.
9. W. Miller, V. Sank, W. Fong, J. Miko, M. Powers, J. Folk, B. Conaway, K. Michael and P.-S. Yeh, " End-to-end communication test on variable length packet structures utilizing AOS testbed," *Proc. SpaceOps'94*, Greenbelt, MD, Nov. 14-18, 1994.
10. L. Miles, J. Venbrux, J. Gambles, J. Hass, W. Smith, G. Maki, S. Whitaker and P.-S. Yeh, "An ultra-low-power, radiation-tolerant data/image compressor for space applications," *Proc. 10th NASA symposium on VLSI design 2002,* Albuquerque, NM., March 2002, pp.9.3.1-9.3.7.
11. P.-S. Yeh, W. Serafino, L. Miles, B. Kobler and D. Menasce, "Implementation of CCSDS lossless data compression in HDF," *Proc. of NASA ESTO Conf. 2002,* Pasadena, CA, June 11-13, 2002.