NICHOLAS D. BESER

SPACE DATA COMPRESSION STANDARDS

Space data compression has been used on deep space missions since the late 1960s. Significant flight history on both lossless and lossy methods exists. NASA proposed a standard in May 1994 that addresses most payload requirements for lossless compression. The Laboratory has also been involved in payloads that employ data compression and in leading the American Institute of Aeronautics and Astronautics standards activities for space data compression. This article details the methods and flight history of both NASA and international space missions that use data compression.

INTRODUCTION

Data compression algorithms encode all redundant information with the minimal number of bits required for reconstruction. Lossless compression will produce mathematically exact copies of the original data. Lossy compression approximates the original data and retains essential information. Compression techniques applied in space payloads change many of the basic design trade-offs that affect mission performance.

The Applied Physics Laboratory has a long and successful history in designing spaceborne compression into payloads and satellites. In the late 1960s, APL payloads used a simple form of data compression to format particle data on the Interplanetary Monitoring Platform (IMP). As the Laboratory prepares to build and launch the Near Earth Asteroid Rendezvous (NEAR) spacecraft, the list of data compression options has expanded to include a draft NASA standard that has had a successful flight history. We have been a contributor to this development through our support of the American Institute of Aeronautics and Astronautics (AIAA) Space Based Observation Systems Committee on Standards. The members of the Space Based Data Compression Standards Panel are the designers of many data compression systems, and two of its NASA participants are the authors of the draft NASA Standard for Lossless Compression.[1] This article will review how data compression has been applied to space missions and show how the methods are now being incorporated into international standards.

Space-based data collection and transmission are basic functional goals in the design and development of a remote sensing satellite. The design of such payloads involves trade-offs among sensor fidelity, onboard storage capacity, data handling systems throughput and flexibility, and the bandwidth and error characteristics of the communications channel. The nature of space probes also requires highly reliable systems that can operate continuously in the presence of multiple component failures. To fulfill these trade-offs and reliability requirements, data compression is applied as an enabling technology.

Performance design trade-offs for a space mission are heavily influenced by size, power, weight, and complexity constraints. As the performance capability changes in a part of the payload chain[2] (Fig. 1), these four basic requirements change. A fifth requirement, mission risk reduction, is always used as the decisive factor. Any decision that increases the risk of incorrectly receiving data is weighed against the benefits of improved performance. In such a trade-off, space data compression has been

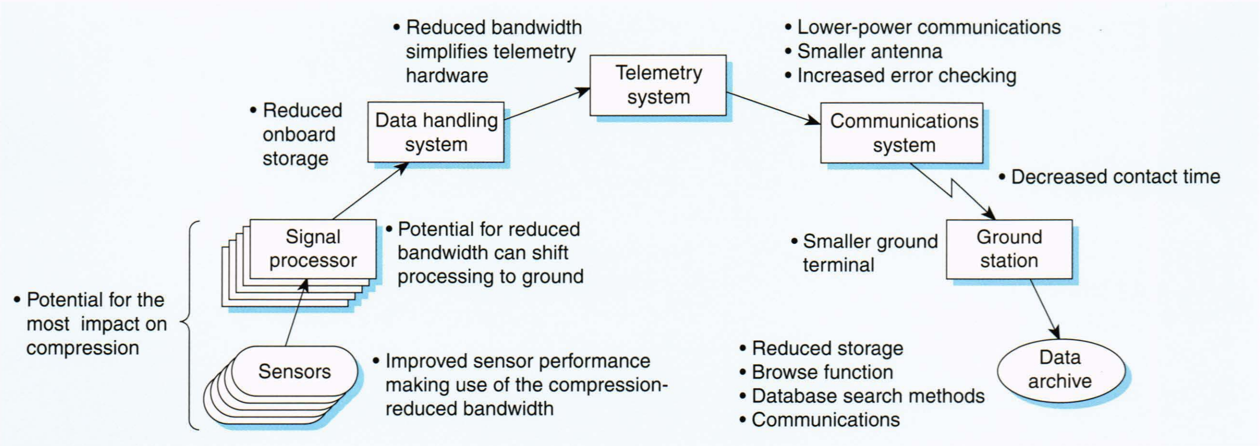


Figure 1. Sensor payload data chain illustrates the trade-offs to be considered when applying data compression in a space mission.

viewed with a sense of duality. Although it makes the mission possible, provides a realistic match between the large range of resolution possible in the instrument and the desired resolution of the mission, and is an established technique that has been used in space since the mid-1960s, it is also sometimes an unnecessary risk factor, and amplifies the effect of communications errors.

As new sensor technology becomes qualified for space use, more capable, higher data volume missions are being proposed. The new sensors will collect large volumes of data while competing for finite communications bandwidths. In addition, the satellites are often forced to share ground station time with other sensor missions. This new technology brings with it the requirement to compress space data while minimizing the risk to the mission.

SPACE DATA STANDARDS

Space-based sensor data tend to be unique for each mission. Data configuration, sensor sensitivity, and sampling rates are dictated by the goals of the satellite. The use of the collected data also differs for each mission. Despite these differences, there is significant interest in using a common approach to data compression. Often, although the data may be different, the techniques are the same. Cost controls on the design process, risk, and reliability suggest that past efforts should be exploited wherever possible. Significant cost savings are also realized when multiple programs design hardware and systems that are interchangeable. All of these factors should be considered when satellite development is limited by both funding and schedule. This awareness is the cornerstone of NASA's "cheaper, better, faster" philosophy.

NASA and the other agencies of the space-faring nations have recognized the need for space data standards by forming the Consultative Committee for Space Data Systems (CCSDS), a voluntary organization that meets periodically to address data systems problems common to all participants and to formulate sound technical solutions to those problems.[1] All committee actions result in recommendations. NASA is in the process of developing a draft standard for data compression that will be submitted to the CCSDS. The draft standard will first become a NASA Goddard standard before being adopted by all of NASA, and then it will be proposed to the CCSDS. One driver behind the draft NASA standard is that many of the techniques used in data compression are common. NASA performed an informal survey of space data compression applications and concluded that 85% of its needs could be met by lossless compression. Standards for compression will also support multimission capability, thus reducing the development cost of subsequent missions. The risk in the use of data compression is also decreased since methods that are well understood and optimized are employed for each mission.

What is Data Compression?

Data items onboard a satellite are routinely collected and transmitted to the ground for later processing. The purpose of data compression is to represent a data item with a smaller number of bits. The reduction results in more efficient use of the finite resources available onboard a satellite. Data compression algorithms reduce the size of the data stream by recognizing redundancy in their mathematical representation and then recoding the data for efficient storage and transmission. The compression algorithms may also remove redundancy by first transforming the data or predicting the value of a data item. The transformed data will recode into a smaller number of bits (see the boxed insert, Coding Methods). Lossless algorithm efficiency is limited by the basic information rate of the data. Algorithms that reduce the information below the Shannon entropy rate of the data incur some loss in data fidelity. Lossy methods will exhibit artifacts depending on the algorithm and how much quantization is used. Quantization is the conversion between high-fidelity and lower-resolution data. The quantization process will produce data with more redundancy and will result in greater compression. Trade-offs among onboard processing, error recovery, and amount of loss result in unique solutions to the data compression problem.

What Data Types are Being Compressed?

Onboard sensor platforms collect a wide variety of data. Table 1 is a partial list of the data types reported in this article; only those programs that have published their methods are included. As seen in Fig. 2, data compression has been used in space since 1967 and is planned for most missions to be launched during this decade.

Data Compression as a Systems Engineering Activity

Systems engineering applied to space missions involves trade-offs that balance the quality of the science data collected against the resources provided by the data handling systems and communications systems. The addition of data compression changes the conditions of the trade-offs so that limited resources that might normally be rejected as not sufficient can be used to perform the mission (Table 2). Data compression can affect all segments of the payload data chain including the sensors, signal processor, data handling system, telemetry system, communications system, and ground station (Fig. 1).

Sensors

Data compression options historically have been part of the sensor design. Decisions on the fidelity of the sensor data, the number of bits, and the scaling of the data are used to decide how much data will be collected to meet the science objectives of the mission. Normally, since mission trade-offs involve decisions on how much data will be sent to onboard signal processors, as well as the statistical characteristics of the data, those decisions are not typically thought of as data compression trade-off issues. They do, however, have implications for the eventual performance of onboard data compressors. When a strict definition of resolution and sampling times is used, the trade-offs fall into the categories of spatial and temporal compression. Averaging adjacent data is an example of spatial compression. Averaging waveforms over time performs temporal compression.

Table 1. Data types that have been compressed onboard satellites.

Data type	Mission	Characteristics
Images	Voyager, Mars Observer, Russian Mars 96, Russian Mars 98, Cassini, Mars Global Surveyor, Clementine, Midcourse Space Experiment (MSX), Galileo	Pixels range from 8–12 bits; data rates range from record playback to real time.
Spectrograph measurements	Mars Observer, MSX, Cassini, Submillimeter Wave Astronomy Satellite (SWAS), Galileo, Russian Mars 96	Spectrograph counts range from 8 bits to over 16 bits; dynamic range depends on integration time and sensitivity of the detector.
Particle counter output	Voyager, Interplanetary Monitoring Platform (IMP), Galileo, Ulysses	Original counts may range up to 32 bits, but may be averaged and then scaled to fit into a small number (8 bits).
Digitized camera output	Radio Astronomy Explorer-2 (RAE-2)	Image is required to indicate antenna position.

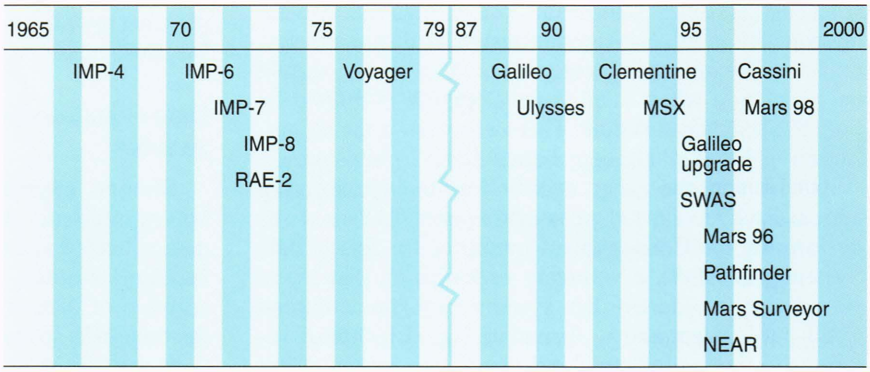


Figure 2. Timeline of missions (past and projected) using onboard data compression.

Signal Processor

Onboard signal processing is used to perform both signal conditioning and postprocessing of the sensor data. Since signal processing involves access to the data and special-purpose hardware, and is typically programmable, the onboard signal processor is an ideal location for performing space-based data compression. After the data leave the signal processor they are routed to the spacecraft data handling system and the telemetry/communications system. Converting the processed sensor data to a compressed data stream in the signal processor is an efficient use of onboard storage and reduces bandwidth requirements for the eventual transmission to the ground station. The onboard signal processor can be augmented with special-purpose compression hardware or firmware containing transformers, quantizers, and encoders. Applying data compression techniques at the signal processing stage is the most efficient approach to the problem of onboard data storage and bandwidth limitations.

Data Handling System

Spacecraft data handling systems are used to temporarily store multiple-sensor data before transmission to the ground through the telemetry/communications system. In general, unless a satellite is engaged in high-priority missions, the data will be stored for relatively short contact periods. Spacecraft designers, when considering the vehicle size, power, and weight, often locate the data compression hardware or software at the centralized data handling system. An additional advantage of this data system is that the sensor output data rates are often much faster than the telemetry and communications processor data rates. This allows more time for the data handling system to perform compression.

Telemetry System

Telemetry systems are nominally sized to support specific data rates driven by both sensor platform needs and the transmitting power of the satellite. In a spacecraft

Table 2. Data compression trade-offs.

Location of compressor	Lossless compression	Lossy compression
Sensors	Requirement for high fidelity with no distortion; real-time processing requirement for compression.	Data editing; data collected with no decrease of scientifically meaningful results; real-time processing requirement for compression.
Signal processor	Signal processor's proximity to raw data typically the first location of data compression; design of the onboard signal processor affected by memory and processing requirements for compression.	High ratio of lossy compression will change the trade-off requirement of where processing is performed.
Data handling system	Improved storage and input/output rates limited by information rate as calculated by Shannon entropy measure.	Improved storage and input/output limited by science value of the decompressed data and could range from 3:1 to 20:1.
Telemetry system	Increased requirement for error-free communications; error containment approach required; rate control approach could change the fixed packet-size feature of telemetry format.	Decreased data rate could allow other instruments larger access to available bandwidth.
Communications system	Error-free communications; moderate savings in bandwidth will permit smaller antenna and lower-power transmitter.	Significant decrease in required bandwidth could affect the size of the antenna and power of the transmitter; change in antenna size can affect decision between fixed and deployable antenna.
Ground station	Moderate reduction in bandwidth.	Major reduction in bandwidth; reduced contact time; smaller archival storage of raw data.

system that employs data compression, telemetry data rates will either be smaller or able to perform the same function with decreased ground station contact time. Unfortunately, an effect of having onboard data compression is a much higher reliance on error-free communications because of the higher risk of data loss. A single bit error on a compressed data stream results in the loss of all undecoded data after the error event. Performance and capability of the telemetry system improve with data compression, even though the requirement of error-free communications implies additional system complexity. The CCSDS has developed a standard that describes the telemetry format and error detection and correction approach. Spacecraft being built since adoption of the standard will normally include concatenated error codes based on convolutional coding and Reed–Solomon codes. The proper use of the CCSDS format will provide a theoretically perfect channel that can support most data compression applications. The CCSDS format also supports the concept of variable packet length messages.

Communications System

Space communications systems are normally a compromise between the amount of data that must be transmitted to the ground and the available bandwidth, power, and contact time allocated to the mission. The performance of the communications channel is measured in terms of the signal-to-noise ratio at a desired frequency and the effective bit error rate in terms of number of bits in error per second. The size of the antenna and the transmitter power affect the performance of a communications channel. In addition, the size of the ground station antenna can improve the effective bit error rate of the system owing to improved signal-to-noise ratio.

Missions designed before the adoption of the CCSDS telemetry format standard used a variety of methods to provide reliable communications, such as higher-power transmission to improve the signal-to-noise ratio, multiple transmission of data, and error detection and correction codes. The CCSDS telemetry format has incorporated error detection bits in the data stream using parameters that can be adjusted for different mission requirements. The communications hardware may also be used to complete the CCSDS format by adding a concatenated code (e.g., convolutional encoding) to reduce the error rates of the channel.

The addition of data compression in a sensor payload changes the performance trade-offs used to select the

CODING METHODS: BACKGROUND FOR LOSSLESS AND LOSSY COMPRESSION

Coding of sensor data in space is an application of Shannon information rate theory (entropy), which, for a source of L independent symbols with a probability p_i, is given by[3]

$$\text{Entropy} = -\sum_{i=0}^{L-1} p_i \log_2 p_i \ \text{ bits per symbol.}$$

For example, the information rate for a uniform distribution of 256 symbols will require exactly 8 bits. Since the data statistics are rarely uniform, the information rate will be smaller than the amount used to store the unencoded data. Coding algorithms use the nonuniform probability distribution to design a new bit encoding that will represent the data at or near the information rate. Original information as shown in Fig. A is measured by entropy. The compression ratio of lossless codes is limited by the basic information rate. Figure B illustrates the development of a simple Huffman code based on a probability distribution.

Coding methods such as run length encoding and Huffman coding represent the data using a variable-length number of bits. Effects of bit errors are amplified since code value and length are defined by a unique pattern of bits. Variable-length codes require an error containment strategy so that the data stream can be recovered if an error occurs.

Codes are designed to optimally describe a unique set of probabilities. Since these probabilities will change as new images and data are processed, the codes must adapt to the new statistics. Codes that are not correctly designed will result in larger than optimal encodings of the data. This implies that a space-based data collection system must use an adaptive code or provide a method of selecting among many predetermined codes. The predetermined codes will not be able to losslessly compress a data stream as efficiently as adaptive coding methods.

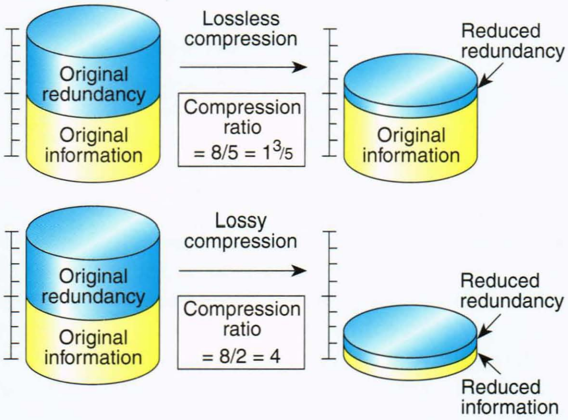
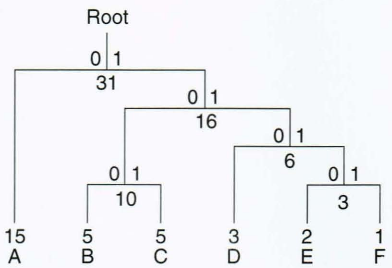


Figure A. Relationship between lossless and lossy compression and the compression ratio.



Symbol	Code
A	0
B	100
C	101
D	110
E	1110
F	1111

Figure B. Example of Huffman code generation. Probability distribution is used to arrange the symbols in terms of decreasing order as leaf nodes on a tree. The two nodes with the smallest probability are merged to form a new node. Codes 1 and 0 are arbitrarily assigned to each pair of branches. Output codes are read sequentially from the root to the leaf node for each symbol.

When the original information is reduced below the entropy level, the compression is no longer reversible. Lossy compression as shown in Fig. A can also be called "approximate compression" since the reconstructed data are an approximation of the original. Data artifacts are produced by lossy compression and can be observed in images or measured mathematically using the following formulas:

Root-mean-squared error (RMSE):

$$\text{RMSE} = \sqrt{1/(NM) \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [D_{\text{input}}(i,j) - D_{\text{output}}(i,j)]^2} \, ,$$

where N and M are the dimensions of the image array and $D(i, j)$ is the value of the original and decompressed data item. Distortion can also be measured as percentage error (PE):

$$\text{PE} = 1/(NM) \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \frac{\left| D_{\text{input}}(i,j) - D_{\text{output}}(i,j) \right|}{D_{\text{input}}(i,j)} \, .$$

final design of the communications system. The more efficient use of the available bandwidth implies that the satellite may be able to use a lower-power transmitter or a smaller antenna (with decrease in weight and fewer attitude control problems owing to the smaller mass of the antenna). Smaller, lower data rates could change the trade-off between a deployable or fixed antenna.

Ground Station

Ground stations provide spacecraft command and control, data capture, data processing, and dissemination. Typically, command and control messages are not a subject for data compression since the command formats are designed for high-reliability transmission and not for smallest possible bandwidth. The other functions of the ground station, however, are directly affected by the use of data compression. Ground stations designed for a satellite system incorporating data compression can receive data at effective rates comparable to higher-power satellite data transmitters with larger antennas. In addition, the routine use of compressed data will allow more satellites to share the use of a ground station, or multiplex the use of a high data rate downlink, since a shorter contact time can be used to receive more data. This becomes an important consideration when the requirement of "better, faster, cheaper" implies that ground stations will not be 100% allocated to a mission, and multiple programs will be sharing the very expensive resources of the ground station, its supporting computer, and its relay satellite network.

The reception of compressed data also affects the ground station storage methodology since much more data can now be handled by the system, providing the data remain in compressed form. Data compression in the data dissemination process can stretch the effective capacity of the media by at least 1.5 times. Data compression is also used to improve the user interaction and postprocessing of the data. By keeping the data in compressed form, the time for retransmitting the data to the user's computer system is reduced. In addition, if the ground station is storing a large amount of data, highly compressed products can be used to provide a "user-defined quality browse." The user can progressively decompress the data so that mission science quality requirements govern the data selection decision while browsing through compressed data sets.

The State of Space-Based Data Compression Systems

Space compression options are limited to the technology that has been flown in space; however, flight-qualified technology is typically several years behind ground technology because of the very conservative nature of space electronics (unusual radiation tolerances) and reliability requirements. Early missions were limited in space-based computing and had low communications bandwidths. The data sent to the ground also normally had a fixed-size value (byte- or word-oriented), and the telemetry systems accommodated them by using fixed-size data packets. Recent missions have used the CCSDS variable-bit-length telemetry format, which also provides a superior error detection and correction scheme. It is possible to design a CCSDS telemetry format so that only one bit error would be expected during the entire mission. Flight-qualified technology has advanced to the level where onboard data compression can be performed either by special electronics or by software using high-speed onboard data processors.

The following section details different data compression techniques selected because of their flight heritage or their significant enhancement to the planned mission. Details of the missions in which they were used are provided to illustrate how data compression has been integrated into the sensor/payload design. Table 3 lists the compression algorithms and the satellite missions.

Table 3. Data compression algorithms flown or planned.

Algorithm	Mission
Predictive	Voyager, Galileo, Mars Observer, Mars Global Surveyor, SWAS, Mars Environmental Survey (MESUR), Mars 96, Mars 98, Cassini
Companding	IMP, Voyager, Galileo, Ulysses, Pathfinder (Mars), MSX, NEAR, Cassini
Data subsampling	RAE-2
Vector quantization	MSX
Transform compression	Clementine, Mars Observer, Mars Global Surveyor, Mars 96, Mars 98, MESUR, Cassini

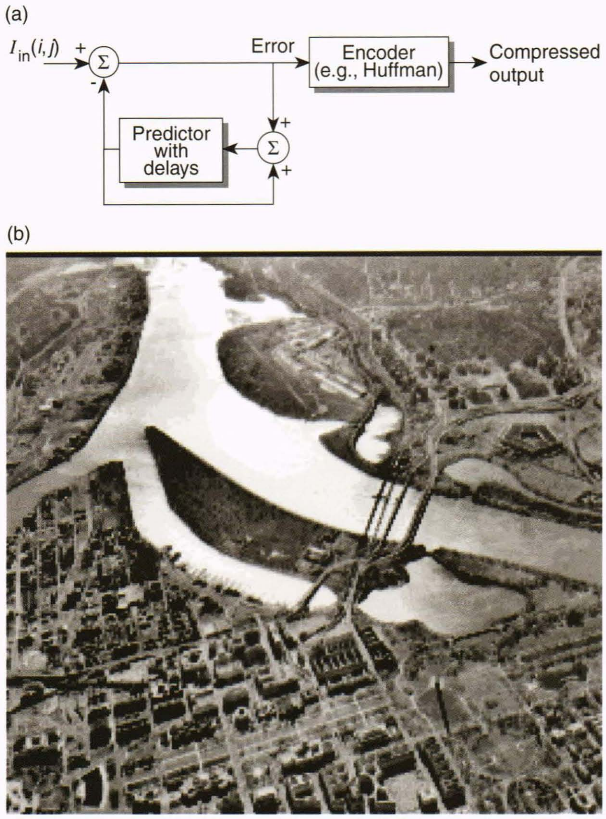
In many of the case studies, compression was not the only option of data transmission. The performance of the data compression algorithm was verified and optimized by sending raw data. Other systems that used lossy compression as an option also provided a data path for lossless or no compression as a method of fine-tuning the quality of the received data. In addition, those systems that supported only lossy compression used two system requirements to justify the selection: the bandwidth was too low to support lossless compression, and the science value was not compromised by distortion contributed by lossy compression.

CODING METHODS FOR SPACE-BASED LOSSLESS AND LOSSY COMPRESSION

Codes are designed to optimally describe a unique set of probabilities. Since these probabilities will change as new images or data are encountered, the codes must adapt to the new statistics. Codes that are not correctly designed will result in larger than optimal encodings of the data. This implies that a space-based data collection system must use an adaptive code or provide a means of selecting among many predetermined codes. The predetermined codes will not be able to losslessly compress a data stream as efficiently as adaptive coding methods.

Predictive Coding

Predictive coding is a method that improves the statistics of the data being coded by removing redundancy between surrounding pixels or data items. The statistics of the resulting sequence will produce a smaller number of bits when represented by a variable-bit-length code. Prediction will depend on statistical characteristics of the data. Waveforms that do not exhibit redundancy for adjacent data points may be redundant if correlated across scans. Design decisions on the predictor will influence the amount of onboard memory and the speed of the signal processor implementing the prediction. As seen in Fig. 3, the predictor design will directly affect the statistics of the predicted error data.[4,5] An ideal predictor will result in near zero error. As can be seen by the formula for entropy (see the boxed insert, Coding Methods), the lossless code for the predicted error will result in a much smaller representation than coding for the raw data.



Original District of Columbia image

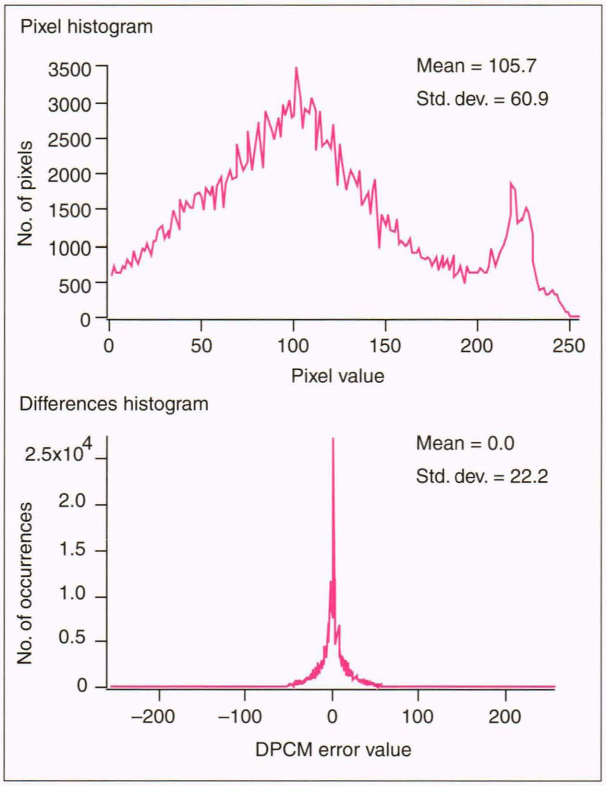


Figure 3. (a) Differential pulse code modulation (DPCM) uses a predictor to improve the probability distribution of the compressed output. ($I_{in}(i, j)$ = image pixel located at row i, column j. (b) Image statistics from the raw data show that a large number of pixel values are used in the range. By using predictors based on nearest-neighbor differencing, the values clustered around zero are most frequent and code to a small number of bits.

Adaptive Coding

Changing data statistics will result in a less than optimal coding performance since each code set is designed against a unique probability distribution. Adaptive coding changes the code to reflect the new distribution of symbols. Collecting and reporting the changed statistics can become a significant overhead factor for space missions since data processor, memory, and communications bandwidth are limiting factors. Robert Rice of the Jet Propulsion Laboratory devised an adaptive code for the Voyager mission,[6–10] which was extremely efficient in terms of detecting changes in data entropy and switching codes without requiring the compressor to maintain and update statistical information. The Rice algorithm is described in the boxed insert, The Rice Encoder Lossless Compression Algorithm. The Rice algorithm is based on a very simple code generator with a programmable data splitter that together comprise a family of code options that are optimal for different entropy ranges. The Rice coders are selected on the basis of a simple estimator of data entropy.

Applications of Lossless Compression in Space Missions—The Rice Algorithm

The Rice algorithm has been used in several space missions (Table 3). Reasons for the success of the Rice algorithm include the following:

1. Lossless compression with very high efficiency. Throughput bench testing of the hardware implementation of the Rice algorithm in the Universal Source Encoder for Space chip has exceeded 25 megasamples (10-bit samples).
2. Low memory requirement. Statistical tables are not required with the Rice algorithm.
3. Can be performed in software using a limited computer. This has been demonstrated on both the Voyager and Galileo missions using the 8-bit 1802 microprocessor.
4. Expandable to higher word size without requiring additional tables or codes.

The Rice algorithm has been proposed as a NASA Standard for Lossless Compression. NASA is also proposing the algorithm as a CCSDS standard[1] and, after acceptance, will be proposed as an International Organization for Standardization (ISO) standard.

The Voyager Mission

The Rice algorithm was first implemented during the Voyager mission, which originally was not planned to go to Uranus and Neptune. After the decision was made to extend the mission, the communications system had to be upgraded because of the decrease in bandwidth caused by the increased distance from Earth. In order to receive data at the extreme distance from Earth, a Reed–Solomon coder was used to improve the bit error rate of the communications system. A lossless compressor was programmed into the flight computer using an early form of the Rice algorithm called the fast compressor, which used a different set of decision tables to select codes where the most significant bits were all 0's. This produced a coder that was about 0.5 bit per sample above the actual entropy. However, the fast compressor could be coded and implemented in the already existing flight architecture of Voyager.

Implementing a variable bit rate coder in a fixed-packet style of telemetry system also presented unique challenges for rate control. A lossless compression algorithm cannot predict how much bandwidth it will use, whereas the Voyager communications system allocated a fixed number of bytes for each transmission. The data compression algorithm compensated for the fixed bandwidth by reading and compressing the data in an alternating reverse scan line order. Figure 4 shows the effect of the reverse scan line on Uranus imagery. As the bandwidth is exhausted, the remainder of the data in the line are not sent. The next transmission will send the reverse scan line data, including all required information to start the decode process, such as the seed value of the differential pulse code modulation. When gaps appear in the images, the adjacent lines can be averaged to recover the information.

The Galileo Mission

Voyager was not the only spacecraft to be retrofitted for data compression. In October 1989, the Galileo satellite was launched using a combination of the shuttle Atlantis and a solid-fuel inertial upper stage. The 4.8-m-high gain antenna did not deploy when it was commanded to open in April 1991. The lack of a means to transmit the sensor results did not mark the end of the mission. Galileo's S-band communications channel was still operational. The communications data rate was reduced from 155 kbits to 10 bits. The spacecraft data handling system included six relatively primitive 8-bit microprocessors (radiation-hard 1802) as well as a tape recorder that could store the data at sensor data rates and play back at the much-reduced rate of the crippled communications system.

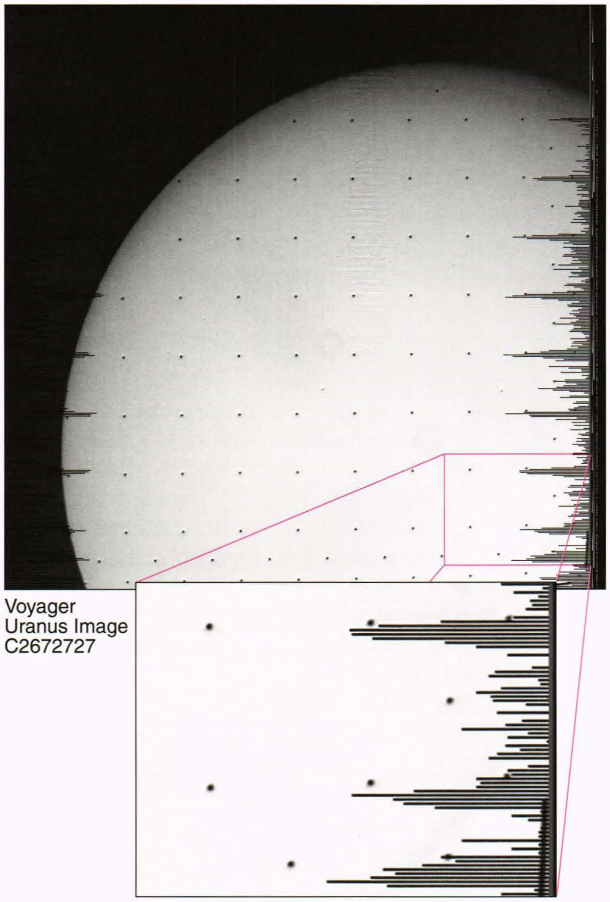


Figure 4. Voyager bit rate control method. When the bandwidth is exhausted by the data from a line, the transmission ends. The next line is read out in a reverse direction so that lost data between lines can be interpolated.

Of the 19 scientific payloads listed in Table 4[11–17], the bulk of the data rate was allocated to the solid-state imager. If the Galileo mission was to continue, all of the instruments would have to transmit the data with a smaller communications bandwidth. In addition, NASA was forced to consider extreme measures to boost the effective data rate of the science telemetry. The error encoding method was modified to improve its recovery characteristics. The ground station antennas were also modified by increasing their size from 64 to 70 m and by developing an array approach to receiving the data. The effective data rate was improved from 10 bits (worst-case estimate) to 100 bits (best-case estimate). NASA also mandated that for the mission to continue, all of the instruments had to compress their data on the average of 10:1. Many of the science payloads had already incorporated data compression into their design. It was necessary for some of the science payloads to reprogram their data collection approach so that the mission science could continue on a much smaller bandwidth. A decision was reached that after the Probe mission, new compression algorithms would be uploaded to Galileo.

Lossless compression has been incorporated into many current and planned missions[18–24] (Table 5). The most common method is the Rice algorithm. Funding problems may cancel some of the planned missions.

Types and Applications of Lossy Compression in Space Missions

Companding

Companding is a mapping function (Fig. 5a), where input data are converted by a lookup table defined by a mathematical function. Since this function maps the original data into a smaller number of bits, it results in a degradation of the original value. The error can be measured against a limiting function. Selection of the mathematical function will depend on the requirements of the instrument and the eventual coding of the output data for telemetry to the ground. This function can be accomplished either as a hardware function (i.e., ROM lookup table) or as a software-implemented lookup table as planned for NEAR and the MESUR Image Processor.

An early example of companding used for onboard data compression occurred with an APL instrument flown on the IMP in 1967. The particle counter used a lossy compression device incorporated with the sensor. Similar approaches were used in other APL sensor missions including Voyager, Ulysses,[24] Galileo,[15] and MSX

Table 4. Galileo antenna recovery data compression.

Experiment	Data type	Compression before antenna problem	Compression after antenna problem
Probe			
Atmospheric structure instrument	Environment parameters	None	None
Neutral mass spectrometer	Spectrograph	None	None
Helium abundance detector	Data readout	None	None
Nephelometer	Particle counts	None	None
Net-flux radiometer	Data readout	None	None
Lighting and energetic particles	Counts	None	None
Spacecraft			
Solid-state imaging	800 × 800 × 8 pixels	Block adaptive rate compressor	Integer cosine transform
Near-infrared mapping spectrometer	Spectrograph	None	Rice
Ultraviolet spectrometer	Spectrograph	None	Rice
Extreme ultraviolet spectrometer	Spectrograph	Small bits	Small bits
Photopolarimeter radiometer	Spectrograph	None	Rice
Magnetometer	Counts	None	Rice
Energetic particles detector	Counts	Log compression	Sampling change/ log compression
Plasma detector	Counts	None	Rice
Plasma wave	Array counts	None	Integer cosine transform
Dust detector	Data points	Encoded data	Encoded data
Radio science (RS)—celestial mechanics	Signal strength	None	None
RS—propagation	Signal strength	None	None
Heavy ion counter	Counts	None	None

Ultraviolet and Visible Imaging and Spectrographic Imaging (UVISI) Sensor.[25] The particle counter scans and sums the counts for a period of time related to the spin rate of the sensor platform and then sends its count to the ground (Fig. 5b). The large number of 24 to 32 bits would be log-compressed to a smaller value of 8. The resulting 8-bit number would represent the magnitude of the counts. Since the counts were very large and exhibited much variation, the magnitude of the counts was the most meaningful for the analysis of charged particles. The mission science was still able to be accomplished with the loss of resolution. As seen in Table 6, several guidelines were selected to decide the mathematical function of the compander. The form of the companding function is derived by selecting the amount of allowable error. Error in images or counts can be measured using the same criteria as described earlier to measure the amount of lossy compression: the mean squared error and the percentage error.

The companding function can also be selected to eliminate noise from the sensor. The MESUR Image Processor sensor converts each pixel using a 12-bit analog-to-digital converter. The effect of sensor noise is eliminated by using a pseudo-square-root function. This results in an 8-bit output pixel, which is further compressed by using lossless predictive coding or lossy transform coding as selected by ground command.

Data Subsampling

Data subsampling illustrates a basic technique of lossy compression. Data are first quantized by deleting part of the data item and then encoding the resulting data for later transmission. The approach of quantization/coding is used in all of the lossy methods that have flown. The Radio Astronomy Explorer-2 (RAE-2) spacecraft[26] used the digital image to visually monitor the four 229-m antenna booms as well as the Moon terminator. Figure 6 details the antenna aspect processor. The RAE-2 compressor was required to perform a 32:1 compression so that the 20,000-bit/s camera could transmit across a 625-bit/s telemetry interface. The compressor would sample every fourth scan line and then use a zero-order predictor and run-length encoding to encode the remaining data. If the information rate was too high for the allocated bit rate, the overflow data would not be sent. This would result in blank data inserted near the end of busy scan lines. The zero-order predictor and run-length encoder produced on the average 8:1 compression. When incorporated with the data sampling by four lines, the effective compression ratio was 32:1.

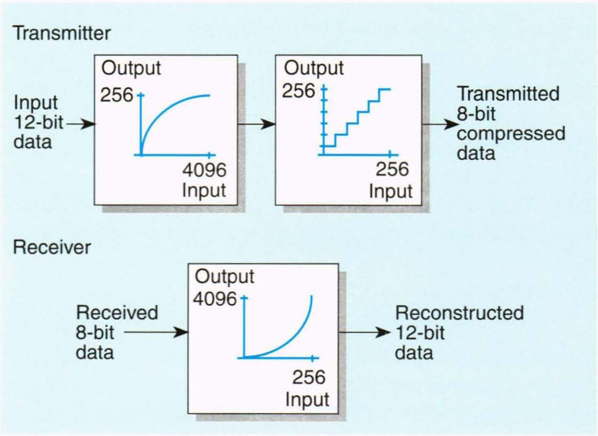
The APL Galileo experiment[15] is also a good example of data sampling compression. The energetic particle detectors (EPD) are collecting particle flux measures as the detector platform spins in space. The instrument will sample for a number of energy ranges and sort the particles by mass, energy, and direction. These measures were normally log-compressed to yield a magnitude number and pseudo-mantissa.

To compress the data from the original 912 bits/s to the average 10:1 compression, several decisions on science value were developed. Of the types of measures the instrument was collecting, direction was determined to be more important than temporal variations. If resources were not a concern, then the EPD instrument could be recoded to do onboard data averaging, editing, and compression. Since storage was at a premium, it was

Table 5. Space lossless compression examples.

Mission	Instrument	Compression
Mars Observer	Mars Observer camera	Hardware differential pulse-code modulation (DPCM) with Huffman coding using fixed Huffman codes
	Gamma-ray spectrometer	Rice algorithm using software-implemented ψ9 coder option
	Thermal emission spectrometer	Rice algorithm implemented in software
Mars Global Surveyor (Mars Observer Recovery Mission)	Mars Observer camera	Hardware DPCM with Huffman coding using fixed Huffman codes
	Gamma-ray spectrometer	Rice algorithm using software-implemented ψ9 coder option
	Thermal emission spectrometer	Rice algorithm implemented in software
SWAS	Acoustic-optical spectrometer	Rice algorithm implemented in software
Mars Pathfinder	MESUR Image Processor	Lossless mode using software implementation of ψ14 Rice algorithm
Cassini	Huygens probe, gas chromatograph, mass spectrometer	Rice algorithm implemented in software
	Huygens Titan descent imager spectral radiometer	Rice algorithm implemented in software, ψ fast for imager and ψ14 for all other data (spectrograph, photometric profiles, etc.)
	High-resolution imaging spectrometer	Rice compression using hardware implementation of ψ14
	Magnetometer	Rice algorithm implemented in software
Russian Mars 94 (now Mars 96)	Automatic solar system particle experiment with rotating analyzer	Rice algorithm implemented in software

(a)



(b)

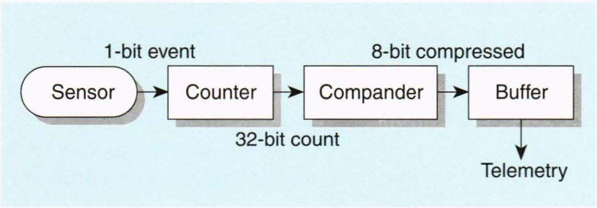
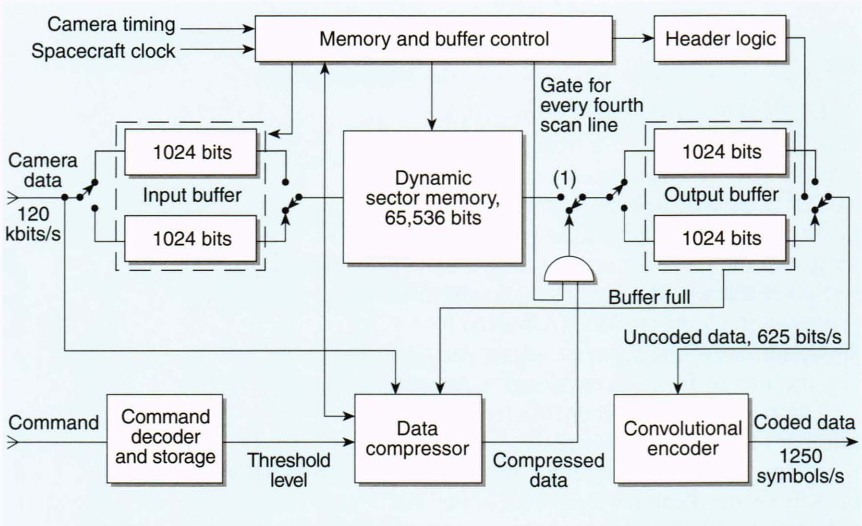


Figure 5. Companding quantization will reduce the resolution of the data without affecting the science value. (a) High-resolution data are quantized by a 12- to 8-bit compander designed to minimize error. The 8-bit data are decompressed into 12 bits by using an inverse function. (b) Companding is used in a sensor payload to adjust for noise or to extract required information such as magnitude of the data.

Table 6. Examples of data compression using companding on space missions.

Mission	Instrument	Companding guidelines
Explorer series, IMP-4, -6, -7, and -8	Science payload—solar particle measurements experiment, charged particle measurements experiment	Mantissa and magnitude (exponent)
Voyager	Low-energy charged particle detector	Mantissa and magnitude (exponent)
Ulysses	Hi-scale sensor	Mantissa and magnitude (exponent)
Galileo	Energetic particle detector	Mantissa and magnitude (exponent)
MSX	UVISI imager	Constrained error (1%)
Pathfinder	MESUR Image Processor (imager)	Pseudo-square-root
NEAR	Imager	Pseudo-square-root
NEAR	X-ray/gamma-ray spectrometer	Constrained error (1%)
Cassini	Particle detector	Mantissa and magnitude (exponent)



(1) Switch in search mode position

Figure 6. Antenna aspect processor samples every fourth scan line and then combines zero-order predictor with run-length encoding. Reprinted from Ref. 26 by permission, © 1976 IEEE.

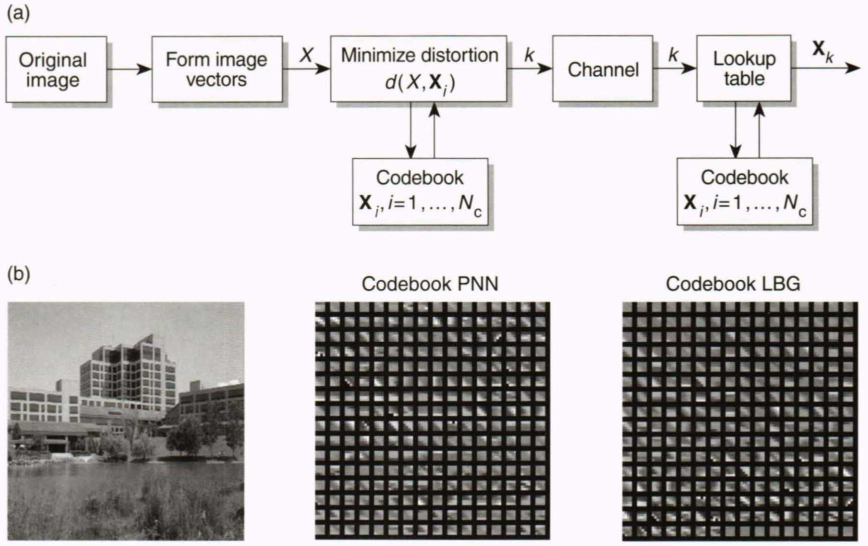
necessary to change the way data were collected in the instrument. Before the antenna problem, data were collected on a time basis. To average adjacent spins, it was necessary to collect data on a spin basis. In addition, the spin data were divided into quadrants. Once the data were available on a spin, they could be averaged with adjacent spins to result in a much smaller bandwidth. Averaging can occur up to 12 min. The 32-bit counts are then companded to yield a 9-bit value. The final output telemetry was command-selectable at 5, 10, 15, 20, 30, 40, and 912 bits/s. The 912 bits/s was the original data rate provided so that high-resolution event monitoring could be performed.

Lossy High Ratio of Compression Systems in Space

Lossy compression of sensor-collected data will result in measurable error. Reasons for selecting lossy compression are normally driven by science, storage, or communications requirements. If the mission science can still be performed in the presence of the errors, then the risk in using lossy compression is acceptable. When the bandwidth or onboard storage is not sufficient to hold or transmit the entire data set, then no mission science will be collected without lossy compression. Space missions during the 1960s and 1970s did not have sufficient onboard processing capability to perform a high ratio of compression with minimal loss. The advent of space-flyable very large scale integrated (VLSI) components has made it possible to use several types of lossy compression techniques for high data rate, high compression ratio applications. Two lossy data compression algorithms that are space-qualified are vector quantization (VQ) and discrete cosine transform (DCT) compression.

Vector Quantization

For VQ, small vectors of data are created by dividing the data set into square regions. The vectors are compared against a database of sample vectors that represent the most common vectors expected to be transmitted (Fig. 7).



National Imagery Transmission Format 13 test image (scaled 0.5)

Codebook based on 4 × 4 vector

Figure 7. (a) Vector quantizer finds the smallest distortion code vector from the collection of codebooks. The index is sent to a receiver that holds the same codebook. (X = square image fragment of the same dimension of the codebook; $\mathbf{X}_i$ = representative image fragment defined by the codebook; N_c = no. of codes in the codebook; k = index of codebook; $\mathbf{X}_k$ = reconstructed image fragment from the kth index in the codebook.) (b) Codebooks are developed from test images using a best-fit algorithm such as the pairwise nearest-neighbor (PNN) or the Linde–Buzo–Gray (LBG).

A database of most common image vectors is developed by training the algorithm using typical data that the sensor will acquire and wish to compress. Several algorithms exist[5,27] that will determine the optimal (most representative) database. Algorithms such as the Linde–Buzo–Gray and the pairwise nearest-neighbor measure the distortion between the selected database and the data training set. Compression is achieved by transmitting the smaller index of the database vector and then reconstructing the image from the received indices.

The quality of the reconstructed data depends on the number of the vectors in the database and the amount of searching that the compressor will perform to find the best-fit vector. Both requirements increase the amount of onboard memory and processing power needed to perform compression. Utah State University developed a unique solution in its design of the MSX space infrared imaging telescope (Spirit III) infrared sensor data compression hardware.[27] The original data are produced at either 16.59 or 3.32 Mbits/s depending on the experiment. As a quick-look and health check function, the infrared data can be sent at real time by a 1-Mbit/s telemetry line. Since the telemetry is shared among different functions, only 667 kbits/s are available for the Spirit III instrument. The Spirit III data compression hardware is designed to compress at ratios of 4:1 and 10:1 by using a combination of proprietary VQ encoder logic and a residual error transmission.

The VQ encoder is based on an algorithm known as mean residual vector quantization (Fig. 8). The mean of each vector is subtracted off and transmitted with the VQ indices. The VQ process is performed using a VLSI chip designed at Utah State University that compares the vector to a database of 256 vectors. The resulting index is then used to generate a residual vector that is sent to the lossless compression hardware for later transmission. Lossless compression is achieved with a constrained, fixed Huffman code. Fixed Huffman tables normally result in larger encoding than the existing original data. A constraint is used to prevent expansion. If the number to be encoded is a low-probability pixel, then a short header code word is sent, followed by the pixel value.

The lossless compression hardware is modified to send as much residual error as desired, so that the quality of the reconstructed data can be constrained not to exceed a maximum residual error. Ratio of compression is controlled by setting the allowable residual error higher or lower. Since the lossless encoder generates a variable bit output code, buffer control to the threshold hardware is used to set the exact compression ratio.

Discrete Cosine Transform Compression

Transform-based methods have been used to perform data compression in software and hardware systems for many years.[4,5] The development of an international standard for still image compression,[4,28] ISO Joint Pictures Experts Group (JPEG), has resulted in a common description of the transform compression algorithm. As a consequence, a number of semiconductor manufacturers have developed hardware that could handle high-speed compression and also be mass-produced at low cost. Some chips were built using components that can operate in space. For the first time, sufficient onboard signal processing capability exists to permit high-quality lossy compression as a sensor postprocessing feature. Advances in the design of high-speed data processing hardware also provide sufficient throughput and memory to support non-real-time, transform-based compression.

Transform-based compression removes spatial redundancy by first performing a spectral decomposition of the data (Fig. 9). The transform coefficients can be combined with a set of basis functions to reconstruct the original data. The coefficients are decorrelated in that most of the information is contained in a few of the coefficients. Many of the coefficients can be deleted without a visually noticeable effect on the reconstructed

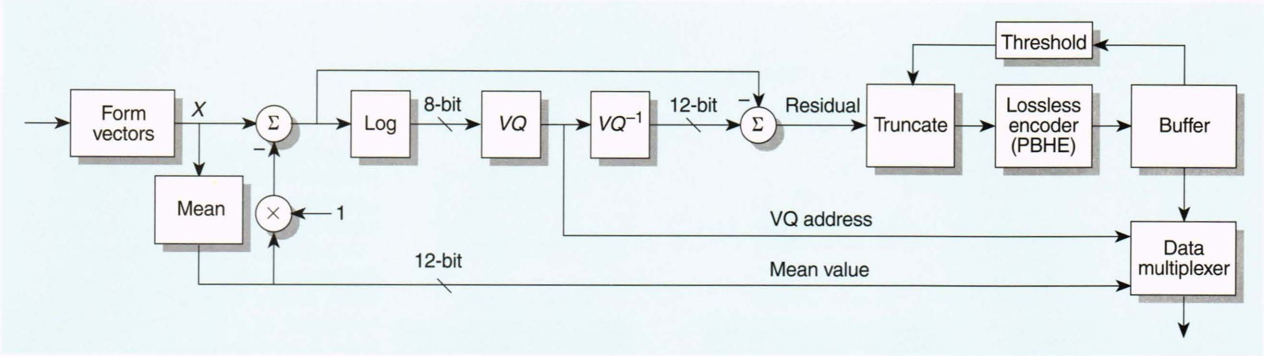


Figure 8. Spirit III vector quantizer is based on the mean residual error approach. The mean of the vector is computed and then subtracted. The 12-bit data are log-compressed to 8 bits prior to VQ. The vector index is decompressed for use by the residual error stage. The residual error is truncated and then encoded using a pixel-based Huffman encoder (PBHE). The rate control algorithm uses the status of the output buffer to set the threshold level used by the truncation hardware. The residual error is then multiplexed with the VQ index and the mean value.

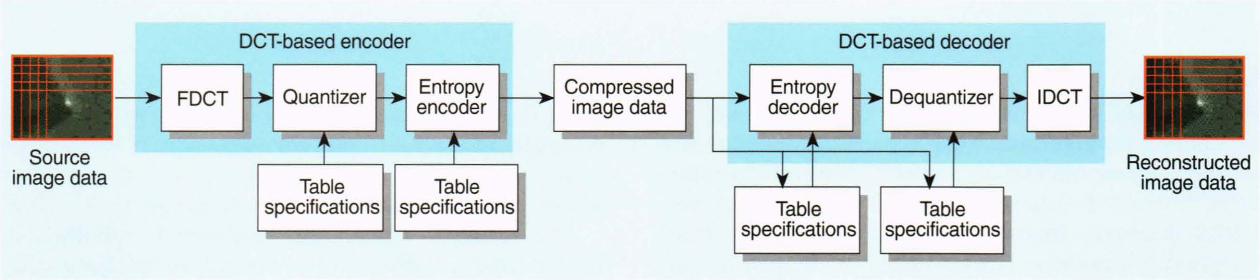


Figure 9. Transform compression relies on the creation of a decorrelated set of coefficients that represent the frequency components of the data. Images are processed with a fast discrete cosine transform (FDCT), and their coefficients are quantized and encoded prior to output on a communications channel. Compressed data are decoded, converted back into transform coefficients, and then processed by an inverse discrete cosine transform (IDCT).

image. The remaining coefficients are losslessly encoded and then transmitted to a receiver where they can be reconstructed into an approximation of the original data.

At least seven missions[19,22,23,29–36] have planned to use or have already flown transform-based compressors (Table 7). Information about the Ballistic Missile Defense Organization Clementine mission and NASA Pathfinder Mars mission has been obtained from viewgraphs of the Clementine critical design review and Pathfinder design implementation and cost review. The Mars Observer camera used a software-implemented compressor that can use either a Walsh–Hadamard or DCT. The onboard processor takes about 30 min to compress a single 2048 × 2048 pixel image. Compression ratios were expected to be up to 10:1, with little visible artifacts. The DCT used a 16 × 16 block size.

The Mars Observer was lost just before reaching orbit around Mars. NASA has proposed to fly selected instruments from the Mars Observer mission on the new Mars Global Surveyor satellite. The Mars Observer camera will be flown again and will use the same DCT method, unless replaced by a newer lossy image compression algorithm.

The remaining five payloads that use lossy compression are unique in that the compression hardware or software handles many channels of image and spectral data (Table 6). Four of the five payloads use the Matra Marconi Space JPEG chip set. Matra Marconi has developed an implementation of the JPEG chip set that is radiation-hard, low in power, and can compress from 3 to 20:1. The hardware module provides an ideal real-time compressor for multichannel imaging sensors. Since the JPEG hardware operates at speeds higher than the individual sensors, it can be used to compress multiple sensor streams for later transmission to the ground.

Lossy compression is a more controversial decision for spaceborne payloads, since the reconstructed data will be approximations of the original. Lossy compression has been judged acceptable[29] for missions that are using the data for photogrammetry, cartography, photoclinometry (shape from shading), spectrophotometry, and photogeology. Lossy compression for low ratios can often provide results with no visual artifacts. Since most of the data collected by space missions will be processed by computer, the artifacts could result in significant loss of information. Figure 10 shows data from the recent Clementine mission. The lunar image does not appear to have any visible artifacts. After performing histogram equalization to enhance fine detail, the 8 × 8 block artifacts become apparent.

Table 7. Transform-based sensor compression missions.

Mission	Payload	Transform type	Number and type of sensors	Compression ratio	Hardware/ software
Mars Observer	Mars Observer camera	Walsh or DCT (16 × 16 pixel block)	1 imager—2048 × 2048 pixels	Up to 10:1	Software
Mars Global Surveyor	Mars Observer camera	Walsh or DCT (16 × 16 pixel block)	1 imager—2048 × 2048 pixels	Up to 10:1	Software
Clementine	Sensor group I and sensor group II	JPEG 8 × 8 pixel block	5 imagers—6 detectors • Star Tracker (2)—384 × 576 pixels • UV/visible—385 × 288 pixels • HiRes—384 × 288 pixels • Near IR—256 × 256 pixels • Long-wave IR—128 × 128 pixels	Nominally 4:1; data content-driven	Hardware (Matra Marconi)
Pathfinder	MESUR Image Processor	JPEG 8 × 8 pixel block	512 × 256 pixels split for stereo and multiple filters	Compression range from 4:1 to 8:1	Software (IBM RISC)
Cassini	Descent imager spectral radiometer	JPEG-like (16 × 16) pixel block	Several imagers with detectors • 512 × 256 pixels • 176 × 256 pixels • 160 × 256 pixels • 128 × 256 pixels	Compression range from 4:1 to 8:1	Hardware (Matra Marconi)
Mars 96 (Russia) (was Mars 94) and Mars 98 (Russia) (was Mars 96)	High-resolution stereo camera (HRSC) and wide-angle optoelectronic stereo scanner (WAOSS)	JPEG 8 × 8 pixel block	High-volume data with resolution at 10 m/pixel • HRSC—5184-pixel swath, 3 focal planes, 9 channels • WAOSS—5184-pixel swath, 1 focal plane, 2 channels	Compression of more than 5:1 required	Hardware (Matra Marconi)

THE NEAR MISSION AND DATA COMPRESSION STANDARDS

Satellite designers can no longer rely on the availability of high-bandwidth communications channels to transmit sensor data. The philosophy of "better, cheaper, faster" will translate into lower power, lower weight, flight heritage, and lower risk. The APL NEAR mission demonstrates these trade-offs. NEAR will rendezvous and orbit an asteroid and send images back to Earth during its 12-month stay. APL is designing, integrating, and launching the satellite in 27 months from project start. Because of the short schedule, the selection of the data compression algorithm has been delayed until other parts of the instrument and data handling system design are complete. Hooks have been made into the design of the flight computer to support compression. The combination of short schedule, desire to minimize risk, and the need to keep the options for compression open until the rest of the payload design has been completed has led to the design decision to perform data compression in software in the instrument processor. The NEAR data communications system already supports the bit error tolerance requirement since it uses the CCSDS standard for telemetry.

The NEAR is a small, low-weight, low-power satellite. The sensor payload will be bandwidth-limited to an average rate of 2900 bits/s when first arriving at Eros to as much as 8000 bits/s later in the mission. Approximately 75% of the available bandwidth is allocated for science data. In addition, the ground station will only receive data 8 h/day. The sensor is based on the Thompson charge-coupled device array and is configured at 244 × 537 pixels with 12-bit resolution. Data are read out once every second, and the derived data rate is 131,028 pixels/s or 1,572,336 bits/s. The combined data rates yield a maximum number of images per day as 39 to 159 images per day transmitted without compression. The overhead bandwidth will be smaller for the higher transmission rate.

Lossless compression can extend the number of images collected to a range of 58 to 238 per day if a 1.5:1 compression ratio is assumed. The actual compression ratio could be much higher depending on the data being

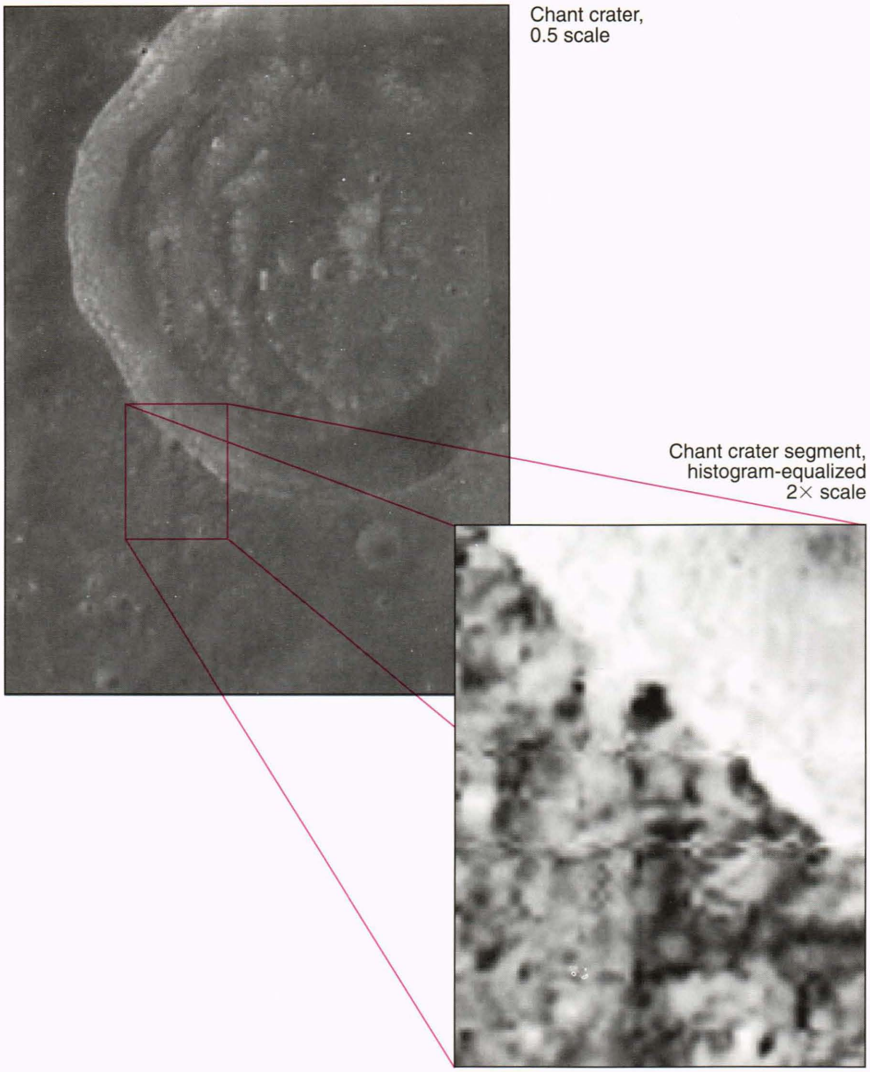


Figure 10. Clementine lunar images illustrate the effect of JPEG lossy compression on image quality. The original image does not have any apparent compression artifacts; however, when the image has been processed with a histogram equalization algorithm, the block artifacts become visible.

compressed. Data compression reduces the risk to the science data since more data can be collected earlier in the mission. Since the data rate is very slow, the instrument data processor can perform onboard data compression with no effect on the hardware design. A software design implies that the code can be designed much later in the schedule, after the long lead time components have been ordered.

CONCLUSION

As seen by the extensive flight history and list of planned missions, data compression can play a major role in the design and successful operation of remote sensing satellites. The trade-offs considered when including data compression provide benefits to all the systems comprising the payload/data chain (Table 2). Compression can be designed into the system without loosing any science value, and it can improve the utilization of precious flight resources. Programs like NEAR are benefiting from data compression's long history of successful use in space. Many of the design issues were captured in the draft NASA standard. Lossless and, to a lesser degree, lossy compression should be considered an enabling technology, helping to make the smaller, lower-cost satellites as productive in data volume as older, larger satellites.

ELECTRONIC MAIL CONTRIBUTORS

Contributions of the AIAA Space Based Data Compression Panel and other researchers are gratefully acknowledged. The Space Based Data Compression Panel is chaired by Nicholas Beser of APL, and Thomas Lynch of Hughes Space and Communications Group. Many members of the panel are the systems engineers, designers, and programmers of the space data compression payloads described in this article. The panel maintains an electronic mailing list hosted by APL which fosters discussion on different data compression topics and is working on the development of space data compression standards. To join the e-mail list, send your e-mail address to space-comp-std-request @aplcomm.jhuapl.edu.

THE RICE ENCODER LOSSLESS COMPRESSION ALGORITHM

The Rice encoder architecture (Fig. A) consists of a preprocessor stage that performs a predictive operation on the original data. The predictor will result in symbols that are centered about zero. The output of the predictor is sent to a mapping function that will convert the signed predicted error symbol into a new positive symbol coded into a standard source δ where

$$\delta = \delta_1, \delta_2, \ldots, \delta_i, \ldots, \delta_j ,$$

where each δ_i, for $1 \le i \le j$, is an N-bit symbol. The property of δ will have the following characteristics:

1. Each δ_i is a non-negative integer.
2. Each δ_i is independent.
3. The probability of occurrence p_i of each δ_i is ordered so that the smaller-value integers occur more frequently. For example, if $p_i = Pr[\delta_j = i]$, where $Pr[]$ is a function that yields the probability that symbol δ_i has value of i, then

$$p_0 \ge p_1 \ge p_2 \ge \ldots$$

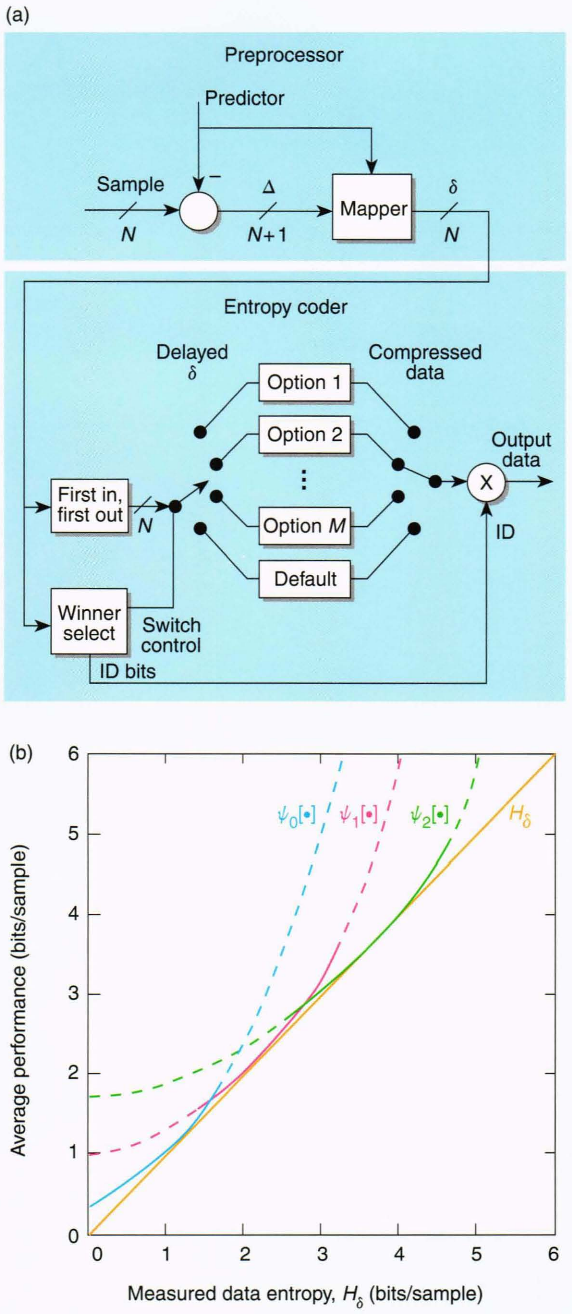


Figure A. Rice adaptive encoder architecture. (a) The sample is preprocessed to remove redundancy in the data and to create a positive set of data. The entropy of the data is estimated and the optimal coder is selected. (Δ = output from DPCM (+ or −); δ = positive-mapped output from DPCM; $\delta = > 0, 2 \times \Delta$; $\delta = < 0, 2 \times |\Delta| - 1$). (b) Each encoder has a specific performance range where its resulting code is close to the entropy of the data. By selecting codes that are close to the optimum performance region of each coder, the Rice algorithm can adapt to changes in data statistics.

The basic Rice coder is called the fundamental sequence (FS). The FS coder simply outputs i 0's for symbol δ_i, followed by a 1. A Huffman code that is generated on the basis of a Laplacian probability distribution will derive to the fundamental sequence.[10] The FS coder has also been called the ψ_1 coder. If the only code available was the FS, the compressed data output would be the concatenated bits

$$\psi_1[\delta] = \psi_1[\delta_1]*\psi_1[\delta_2]*\ldots*\psi_1[\delta_j] ,$$

where * means concatenation. The ψ_1 coder is optimum between 1.5 to 2.5 bits per sample. Other entropy ranges are accommodated by devising a sample-split mode, where the least significant bits of the sample are considered random and are sent without coding. The most significant bits are encoded with the FS coder. The split-sample data item consists of a random component (the least significant bits) and a component that follows a Laplacian distribution (the most significant bits). (See Fig. B).

Each data item is represented by $\delta_i = m_j*lsb_j$, where m_j is the most significant bits and lsb_j is the least significant bits. The most significant bits can be concatenated together as $\tilde{M}^{n,k} = m_1*m_2* \ldots *m_j$. The least significant bits can be concatenated together as $\tilde{L}_k = lsb_1*lsb_2* \ldots *lsb_j$. The ψ_1 coder applied as $\psi_1[\tilde{M}^{n,k}]*\tilde{L}_k$ defines the $\psi_{1,k}$ coder. There are two special cases for the split-sample coder:

$$k = 0 \Rightarrow \psi_{1,0}[\delta] = \psi_1[\delta]$$

and

$$k = n \Rightarrow \psi_{1,n}[\delta] = \delta .$$

The $\psi_{1,k}$ coder operates on small blocks of data so that as the statistics of the data change a new coder will be selected. The performance of the Rice algorithm depends on the entropy estimate of the data. The performance of the coder can be estimated by summing the data values and then comparing against limits that are functions of the size of the data blocks. The table describes the decision regions for 8-bit data, with J data items per block. There are eight Rice coder options in the table, which would be selected with a 3-bit identification (ID) code. Expanded versions of the $\psi_{1,k}$ coder will support data rates that range from 1.5 up to 15 bits per pixel (for the $k = 14$ case). Figure C shows an example of a Rice encoder using a data fragment from the same data set used in Fig. B in the boxed insert, Coding Methods. The Rice algorithm will adapt to changing statistics of the data, whereas the Huffman codes are redefined to accommodate new distributions of the data.

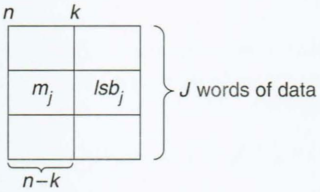


Figure B. Block organization for split-sample mode. Least significant bits are concatenated without compression.

Data rates below 1.5 bits per pixel are handled by two low-entropy options. The zero-run option encodes the number of zero blocks of concatenated difference values as a fundamental sequence. A limit of 63 runs of zero blocks can be coded. A special end-of-line code is used to signal the condition when the remaining blocks of a specified packet have zero values and the number of zero-run blocks is greater than 4.

The extended low-entropy option is used when the mapped symbol differences have an entropy value near zero. The encoding scheme pairs every two mapped symbols before recoding them with the FS coder. The procedure for the extended low-entropy suboption is given in Ref. 1.

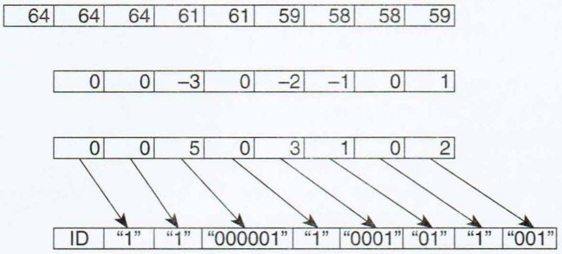


Figure C. Data are processed by a differential pulse code modulation algorithm (top). Signed values are mapped to all positive, and then summed to select the Rice coder (center). Sum of mapped values = 11, with a block size of 8, selects the $\psi_{1,0}$ coder as shown by the table in this boxed insert.Total length of the output code is 19 bits plus ID code (bottom).

Code selection regions based on the sum of data items. J is the number of pixels in a group to be coded, and F_0 is the sum of the pixels in a group. Eight-bit data are assumed.

Code option	F_0 region (bits)
$\psi_{1,0}$	$F_0 \le 5(J/2)$
$\psi_{1,1}$	$5(J/2) < F_0 \le 9(J/2)$
$\psi_{1,2}$	$9(J/2) < F_0 \le 17(J/2)$
$\psi_{1,3}$	$17(J/2) < F_0 \le 33(J/2)$
$\psi_{1,4}$	$33(J/2) < F_0 \le 65(J/2)$
$\psi_{1,5}$	$65(J/2) < F_0 \le 129(J/2)$
$\psi_{1,6}$	$129(J/2) < F_0 \le (128n - 831)(J/2)$
$\psi_{1,7}$	$(128n - 831)(J/2) < F_0$

REFERENCES

[1] *Source Coding For Data Compression*, Consultative Committee for Space Data Systems Recommendation for Space Data Systems Standards, CCSDS 111.0-W-2 White Book, CCSDS Secretariat, NASA Headquarters, Washington, DC (Mar 1994).

[2] Beser, N., "Development of a AIAA Standard for Space Based Data Compression," in *Proc. Computing in Aerospace 9 Conf.*, pp. 129–136 (Oct 1993).

[3] Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ (1989).

[4] Pennebaker, W. B., and Mitchell, J. L., *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York (1993).

[5] Rabbani, M., and Jones, P. W., *Digital Image Compression Techniques*, Vol. TT7, SPIE Press, Bellingham, WA (1991).

[6] Rice, R. F., and Lee, J. J., *Some Practical Universal Noiseless Coding Techniques, Part II*, JPL Publication 83-17, Jet Propulsion Laboratory, Pasadena, CA (1 Mar 1983).

[7] Rice, R. F., *Some Practical Universal Noiseless Coding Techniques*, JPL Publication 79-22, Jet Propulsion Laboratory, Pasadena, CA (15 Mar 1979).

[8] Rice, R. F., *End-to-End Imaging Information Rate Advantages of Various Alternative Communication Systems*, JPL Publication 82-61, Jet Propulsion Laboratory, Pasadena, CA (1 Sep 1982).

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

[10] Yeh, P. S., Rice, R. F., and Miller, W., *On the Optimality of Code Options for a Universal Noiseless Coder*, JPL Publication 91-2 (15 Feb 1991).

[11] Belton, M. J. S., Klassen, K. P., Clary, M. C., Anderson, J. L., Anger, C. D., et al., "The Galileo Solid-State Imaging Experiment," *Space Sci. Rev.* **60**, 413–455 (1992).

[12] Carlson, R. W., Weissman, P. R., Smythe, W. D., and Mahoney, J. C., "Near-Infrared Mapping Spectrometer Experiment on Galileo," *Space Sci. Rev.* **60**, 457–502 (1992).

[13] Cheung, K. M., Tong, K., "Proposed Data Compression Schemes for the Galileo S-Band Contigency Mission," *1993 Space and Earth Science Data Compression Workshop*, NASA Conf. Pub. (1993).

[14] Garrard, T. L., Gehrels, N., and Stone, E. C., "The Galileo Heavy Element Monitor," *Space Sci. Rev.* **60**, 305–315 (1992).

[15] Jaskulek, S., *EPD SBand Mission Software Requirements, Revision C*, JHU/APL SII-2-949-B (2 May 1994).

[16] Johnson, T. V., Yeates, C. M., and Young, R., "Space Science Reviews Volume on Galileo Mission Overview," *Space Sci. Rev.* **60**, 3–21 (1992).

[17] Lai, K. M. T., *GCMS Data Compression Scheme*, Hughes STX Design Note (10 Dec 1992).

[18] Russell, E. E., Brown, F. G., Chandos, R. A., Fincher, W. C., Kubel, L. F., et al., "Galileo Photopolarimeter/Radiometer Experiment," *Space Sci. Rev.* **60**, 531–563 (1992).

[19] Albee, A. L., Arvidson, R. E., and Palluconi, F. D., "Mars Observer Mission," *J. Geophys. Res.* **97**(E5), 7665–7680 (25 May 1992).

[20] Boynton, W. V., Trombka, J. I., Feldman, W. C., Arnold, J. R., Englert, P. A. J., et al., "Science Applications of the Mars Observer Gamma Ray Spectrometer," *J. Geophys. Res.* **97**(E5), 7681–7698 (25 May 1992).

[21] Christensen, P. R., Anderson, D. L., Chase, S. C., Clark, R. N., Kieffer, H. H., et al., "Thermal Emission Spectrometer Experiment: Mars Observer Mission," *J. Geophys. Res.* **97**(E5), 7719–7734 (25 May 1992).

[22] Malin, M. C., Danielson, G. E., Ingersoll, A. P., Masursky, H., Veverka, J., et al., "Mars Observer Camera," *J. Geophys. Res.* **97**(E5), 7699–7718 (25 May 1992).

[23] Malin, M. C., Danielson, G. E., Ravine, M. A., and Soulanille, T. A., "Design and Development of the Mars Observer Camera," *Int. J. Imag. Sys. Tech.* **3**, 76–91 (1991).

[24] Lanzerotti, L. J., Gold, R. E., Anderson, K. A., Armstrong, T. P., Lin, R. P., et al., *The ISPM Experiment for Spectral, Composition, and Anisotropy Measurements of Charged Particles at Low Energies*, ESA Special Publication SP-1050, pp. 143–153 (Jul 1983).

[25] Carbary, E. H., Darlington, E. H., Harris, T. J., McEvaddy, P. J., Mayr, M. J., et al., "Ultraviolet and Visible Imaging and Spectrographic Imaging (UVISI) Instrument," *Appl. Opt.* **33**(19) (Jul 1994).

[26] Miller, W. H., and Lynch, T. J., "On-Board Image Compression for the RAE Lunar Mission," *IEEE Trans. Aerosp. Electr. Sys.* **AES-12**(3), 327–335 (May 1976).

[27] Budge, S. E., "The SPIRIT III Radiometer Data Compression System," in *Proc. Computing in Aerospace 9 Conf.*, pp. 622–630, (Oct 1993).

[28] Wallace, G., "The JPEG Still Picture Compression Standard," *Communications of the ACM* (Apr 1991).

[29] Albertz, J., Scholten, F., Ebner, H., Heipke, C., and Neukum, G., "The Camera Experiments HRSC and WAOSS on the Mars 94 Mission," *Int. Arch. Photogram. Rem. Sens.* **29**, Part I, 130–137 (1992).

[30] Cheung, K. M., *Comments on Professor J. Blamon's Viewgraphs "A JPEG Digital Image Compression Module for Space Applications,"* JPL Memorandum 331-92.2-119 (21 Sep 1992).

[31] Michalik, H., Ruffer, P., and Gliem, F., "Versatile On-Line Image Data Compression for Spaceborne Applications," *IEEE IGARSS '91 Proc.*, pp. 1443–1447 (1991).

[32] Morrison, D. A., *The Lunar Scout Program, An International Program to Survey the Moon from Orbit for Geochemistry, Mineralogy, Imagery, Geodesy, and Gravity*, NASA Technical Memorandum 104791 (Apr 1994).

[33] Oertel, D., Jahn, H., Sandau, R., Walter, I., Driescher, H., et al., "MUSIC: Multifunctional Stereo Imaging Camera System for Wide Angle and High Resolution Stereo and Color Observations on the Mars-94 Mission," in *Proc. 41st Congress of the International Astronautical Federation*, Dresden, Germany, pp. 1–7 (6–12 Oct 1990).

[34] Oertel, D., Sandau, R., Deriescher, H., and Walter, I., "Synoptic Wide-Angle Stereo Scanner for the Mars-94 Mission," *Jena Rev.* **3**, 120–123 (1990).

[35] Ruffer, P., Rabe. F., and Gliem, F., "A DCT Image Data Processor for Use on the Huygens Titan Probe," *IEEE IGARSS '92 Proc.*, pp. 678–680 (1992).

[36] Ruffer, P., Rabe, F., Gliem, F., and Keller, H. U., "Comparison of the Lossy Image Data Compressions for the MESUR Pathfinder and for the Huygens Titan Probe," *1994 Space and Earth Science Data Compression Workshop*, NASA Conf. Pub. 3255 (1994).

NOTE: This paper may become part of a forthcoming Van Nostrand-Reinhold book being prepared by N. Beser entitled *Introduction to Data Compression Standards.*

ACKNOWLEDGMENTS: My thanks to Mike Caplinger—Malin Space Associates; Kar-Ming Cheung—Jet Propulsion Laboratory; Lyn Doose—University of Arizona; Andy Feldstein—Innovative Concepts, Inc.; Dave Gingerich—Martin Marietta; Keith Lai—Hughes STX; Erick Malaret—Hughes STX; Jouni Ryno—Finnish Meteorological Institute; Robert Rice—Jet Propulsion Laboratory; Gerald Schlotzhauer—DLR-Institut fuer Weltraumsensorik; Gottfried Schwarz—German Aerospace Research Establishment; and Pen Shu Yeh—NASA Goddard.

THE AUTHOR

NICHOLAS D. BESER received a B.E.S. from The Johns Hopkins University in 1974 and an M.S.E. in solid state physics from The Johns Hopkins University in 1974. He received his Ph.D. from the Moore School of the University of Pennsylvania in computer science and systems engineering in 1983. Before joining APL in 1989, he was a technical staff member at General Electric Space Division where he was responsible for research into space-based data compression. At APL Dr. Beser is a Senior Staff member in the Space Systems Branch of the Space Department, where he specializes in military and space data compression technology. He teaches graduate classes in the Computer Science Department of The Johns Hopkins University G.W.C. Whiting School of Engineering.