13.6 FLEXIBILITY AND ADAPTABILITY OF NOAA'S LOW RATE INFORMATION TRANSMISSION SYSTEM

Jeffrey A. Manning, Science and Technology Corporation, Suitland, MD *

Raymond Luczak, Computer Sciences Corporation, Suitland, MD

Marlin O. Perkins, National Oceanic and Atmospheric Administration, Suitland, MD

I. Introduction

The National Oceanic and Atmospheric Administration's National Environmental Satellite Data and Information Service (NOAA/NESDIS), has begun implementation of the Low Rate Information Transmission (LRIT) Processing System. The LRIT system, which is being developed and implemented under the Computer Sciences Corporation's (CSC) Central Satellite Data Processing Center contract, is being phased in as a replacement for the current Weather Facsimile (WEFAX) Processing System.

NOAA/NESDIS has been distributing products via WEFAX, an analog meteorological satellite broadcast service, since 1976. The image products are in grey scale and are currently being disseminated to direct broadcast users via the GOES L-band downlink frequency (1691.0 MHz). In an effort to provide better service to the user community, and in response to the World Meteorological Organization's (WMO) recommendation for digital meteorological satellite broadcasts, NOAA is implementing a transition from its WEFAX service to the digital LRIT service. The digital LRIT service provides NOAA a level of flexibility and adaptability that it does not currently have with WEFAX, as such, this paper will focus on the benefits of various approaches to compression of digital data.

II. LRIT Specification

While the primary purpose of the LRIT system is to replace the current WEFAX service, an additional benefit of its implementation is that it provides NOAA an opportunity to exploit the characteristics of digital data.

Because the LRIT system utilizes digital data rather than analog data, the data are relatively easy to store and can be altered electronically.

This permits the data to be manipulated in various ways, which allows the data to be subject to compression, segmentation, and encryption. This paper will focus on various methods for compressing data, such as JPEG, JPEG2000, Rice, and Zip.

*Corresponding author address: Jeffrey Manning, STC, FB-4, Rm. 0343; 4401 Suitland Road, Suitland, MD 20746-4303
Email: Jeff.Manning@noaa.gov

The flexibility of the LRIT specification allows the incorporation of various types of mission specific headers. These headers can be used in many ways, including the ability to define various types of data compression. The LRIT system currently applies compression only to image data. The options for compression include:

- No Compression
- Lossless Compression
- Lossy Compression

The LRIT system, for lossy compression, is currently utilizing Joint Photographic Experts Group (JPEG) compression as described in the Coordination Group for Meteorological Satellites (CGMS) LRIT/HRIT Global Specification (Reference 1). In addition to the lossy JPEG compression being used, LRIT has employed the use of lossless Rice compression (Reference 2), recommended by the Consultative Committee for Space Data Systems (CCSDS). There are many more compression algorithms that could be utilized and, as compression algorithms are evaluated and/or new compression algorithms become available, the LRIT development team will make every effort to incorporate them into a better LRIT product for the user community. Compression algorithms currently being evaluated for use in LRIT include JPEG2000, and ZIP compression.

III. The Use of LRIT Headers

The LRIT file format includes header records and a data field record. A level of flexibility is obtained by the LRIT system through the use of various header records. The LRIT system includes mandatory and non-mandatory headers, as well as the opportunity to make use of mission specific header records. Mission specific header records are headers that the CGMS specification allows, but which are not predefined. The mission specific header(s) are self defined by the entity or organization that produces the LRIT product, and define a particular set of parameters that serve a specific function for understanding by that organization. The NOAA LRIT system, for image compression, uses two header records to define compression. The first header, the Image Structure header, is a mandatory header for image data specified in the CGMS specification. The second header, the NOAA LRIT header, is a mission specific header developed to allow NOAA the ability to more finely define the product and the compression being used, as well as the ability to designate compression for non-image data. The Image Structure header determines the structure of the image and is laid out as follows:

Size (octets)	Type	Contents	Value
1	Uint	Header type	1
2	Uint	Header record length	9
1	Uint	Bits per pixel	NB
2	Uint	Columns in image	NC
2	Uint	Lines in image	NL
1	Uint	Compression flag	0 = none 1 = lossless 2 = lossy

Table 1

The NOAA LRIT header determines product identification and also provides specifics as to which scheme is used to compress the data field. The structure of the NOAA LRIT header is as follows:

Size (octets)	Type	Contents	Value
1	Uint	Header type	129
2	Uint	Header record length	14
4	Char	Agency Signature	NOAA
2	Uint	Product ID	0 – 65000
2	Uint	Sub ID	0 – 65000
2	Uint	Parameter field	0 – 65000
1	Uint	NOAA Compression flag	0 = none 1 = Rice 2 = JPEG

Table 2

For the NOAA LRIT system, the compression flag from the Image Structure Header and the NOAA LRIT Header will be required to determine the image compression. For example, if Rice compression is to be used, the lossless compression flag will be set in the Image Structure Header, and the compression field in the NOAA LRIT Header will be set to 1. The information then is in place for the LRIT User Station to determine the compression performed on that file, so that decompression can correctly be applied. The values for the compression field of the NOAA LRIT header can be found in table 3.

In the case of non-image data (e.g. text), the Image Structure header will not exist for the LRIT file generated. The NOAA LRIT header will solely be used to determine which compression scheme, if any, was applied to the non-image data. At the LRIT User Receive Station, the LRIT file will contain the necessary information to determine if compression was applied, and whether to image or non-image data. The NOAA LRIT header compression field will be accessed to determine if the non-image data was compressed and which scheme was applied. The flexibility of using mission specific headers provides the LRIT system the opportunity to utilize compression for more than for image data.

Global Image Compression Flag	NOAA Compression Flag	Compression
0 – None	0 – None	None
1 – Lossless	1 – Rice 2 – JPEG 3 – JPEG 2000 . Lossless compression schemes	Rice compression JPEG compression JPEG2000 compression .
2 – Lossy	Lossy compression schemes	Global-compatible JPEG lossy image compression
None	10 – Zip	Zip compression (Text)

Table 3

IV. Benefits of Compression

Due to the limited bandwidth (128 kb/sec) for LRIT, compression is being employed to maximize the amount of data transmitted, as well as to reduce the latency of the data. Compression of the Data Field record, for an LRIT file, is comprised of three distinct categories. The first category is lossy file compression. Lossy file compression provides for very good compression ratios, but causes a loss of some of the source information content so the data cannot be fully restored to its original content upon decompression. JPEG (ISO standard 10918) is being utilized by the LRIT system per the CGMS specification. The second category is lossless file compression. Lossless file compression does not usually attain the high compression ratios of lossy compression, but has the advantage of being able to decompress the data so that the file structure is exactly as it existed prior to compression. An example of lossless file compression is JPEG2000. Both compression categories addressed thus far involve compression schemes that are applied to a complete file, that is, the file as a whole is manipulated according to the specified algorithms to reduce the data volume of the original file. The use of file compression on the Data Field record creates LRIT files that are greatly reduced in size from the original file structure, but incur a problem if any data from the file is lost or corrupted in the transmission process. The decompression algorithm that must be applied to recreate the original file structure cannot be performed if any portion of the file has been rendered unusable, because the algorithm requires all the compressed file information to restore the file to its original file structure. The compression algorithm, for file compression, is applied to the whole of the original file structure, and as such, the decompression algorithm must also be applied to the compressed file structure, as a whole, to reconstruct the original file structure. This brings us to the third compression category used in the LRIT system, packet compression. Packet compression involves compressing the original source data into a variable length data structure that is then packetized. The packet compression scheme used in the LRIT system is the Rice algorithm. The benefit of using the packet compression scheme is achieved, not only through the data compression, but also in the ability to reconstruct the file structure if any packet is lost or corrupted in transmission. The transmitted data received by the LRIT User Station extracts the contents of the packets and the data units are then provided to the Rice decompression software. Packets that may have been lost or are deemed unusable are replaced with missing values. Using the packet compression scheme allows the compressed data to be reconstructed, even if packets are lost or corrupted. The reconstructed file will contain all data that was received, and if any packets were found unusable, the reconstructed file will show missing values in those locations in the file, but the remainder of the file will contain the data from the original file structure.

Each category referenced above offers a benefit that provides the LRIT system various means to maximize the use of the finite bandwidth available. The combination of the various approaches to compression gives the LRIT system a sense of flexibility that cannot be realized in the current WEFAX system.

V. JPEG Compression

The JPEG compression standard (Reference 3) takes advantage of the limitations of the human visual system and throws away some information, which the eye otherwise, would not easily distinguish. This approach helps to achieve better compression, but creates a result where the original and the decompressed files are no longer the same. JPEG compression works best on natural scenes, where the loss of some information does not distract from the overall quality of the image. The technique used is thus lossy, and information is lost from the original. The key to making JPEG work is choosing what data is to be thrown away.

The method for compressing data in the JPEG process involves dividing the image up into blocks and calculating the discrete cosine transform (DCT) of each block. The blocks are comprised of an array consisting of 8 by 8 pixels. The DCT separates the image into parts, spectral sub-bands, of differing importance. The DCT transforms the image from the spatial domain to the frequency domain. With an input image, A, the coefficients for the output "image," B, are:

$$B(k_1,k_2) = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} 4\cdot A(i,j)\cdot\cos\left[\frac{\pi\cdot k_1}{2\cdot N_1}\cdot(2\cdot i+1)\right]\cdot\cos\left[\frac{\pi\cdot k_2}{2\cdot N_2}\cdot(2\cdot j+1)\right]$$

The input image is N2 pixels wide by N1 pixels high where A(I,J) is the intensity of the pixel in row I and column J, and B(k1,k2) is the DCT coefficient in row k1 and column k2 of the DCT matrix. The DCT input is an 8 by 8 array of integers containing each pixel's gray scale value. For grey scale, 8 bit pixels provide values from 0 to 255. The output array of DCT coefficients contains integers, which can range from -1024 to 1023. For most images, much of the signal energy lies at low frequencies that appear in the upper left corner of the DCT. The lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion.

The next step in the JPEG compression process involves a quantizer, which rounds off the DCT coefficients. This is the step that produces the lossy nature of JPEG, but also allows for the ability to obtain large compression ratios. There is a tradeoff between image quality and the degree of quantization. A large quantization step size can produce unacceptably large image distortion. Unfortunately, finer quantization leads to lower compression ratios. The question is how to quantize the DCT coefficients most efficiently. Because of human eyesight's natural high frequency roll-off, these frequencies play a less important role than low frequencies. This lets JPEG use a much higher step size for the high frequency coefficients, with little noticeable image deterioration. The quantization is performed using a quantization matrix that is an 8 by 8 matrix of step sizes (quanta), containing one element for each DCT coefficient. The matrix is usually symmetric. Step sizes will be small in the upper left (low frequencies), and large in the upper right (high frequencies); a step size of 1 is the most precise. The quantizer divides the DCT coefficient by its corresponding quantum, and then rounds to the nearest integer. Large quanta drive small coefficients down to zero. The quantization results in many high frequency coefficients becoming zero, and therefore easier to code. The low frequency coefficients undergo only minor adjustments. After quantization, it is not unusual for more than half of the DCT coefficients to equal zero. JPEG incorporates run-length coding to take advantage of this. For each non-zero DCT coefficient, JPEG records the number of zeros that preceded the number, the number of bits needed to represent the number's amplitude, and the amplitude itself. To consolidate the runs of zeros, JPEG processes DCT coefficients in a zigzag pattern. The number of previous zeros and the bits needed for the current number's amplitude form a pair. Each pair has its own code word, assigned through a variable length code (e.g. Huffman). JPEG outputs the code word of the pair, and then the codeword for the coefficient's amplitude (also a variable length code). After each block, JPEG writes a unique end-of-block sequence to the output stream, and moves to the next block. When finished with all the blocks, an end-of-file marker is written and the file has been compressed.

VI. JPEG2000

JPEG2000 is the next ISO/ITU-T standard for still image coding. JPEG2000 has the benefit of performing either lossless or lossy compression. The following description (Reference 4) will keep its scope to Part I of the standard, which defines the core system. JPEG2000 is based on the discrete wavelet transform (DWT), scalar quantization, arithmetic coding and post-compression rate allocation. The DWT can be performed with either a reversible filter, which provides for lossless coding, or an irreversible filter, which provides for higher compression but does allow for lossless compression. The quantizer follows and is independent for each sub-band. Each sub-band is divided into blocks, typically 64x64, and entropy coded using context modeling and bit-plane arithmetic coding. The coded data is organized into layers (quality levels) and then are subject to the post-compression rate allocation before being output to the code-stream in packets.

Preprocessing is the first step performed in the JPEG2000 compression process and involves breaking the image into components. The image components are then broken down into rectangular tiles, which the compression is applied to, as if they were distinct images. Following the breakdown of the image into tiles, the wavelet transform is performed. The wavelet transform (DWT) is used to analyze the tile components into different decomposition levels. The decomposition levels are made up of a number of subbands, which consist of coefficients that detail the horizontal and vertical spatial frequency characteristics of the original component. After transformation, all coefficients are quantized. Quantization is the process by which the coefficients are reduced in precision. This is the step in processing that lossless of lossy compression is determined. Compression is lossy unless the quantization step is one (1), and the coefficients are integers. After quantization, each subband is divided into rectangular blocks, which are further divided into non-overlapping rectangles referred to as code blocks. Each code block is the basis for a separate bit stream, in which no information is used from other blocks. The entropy encoding is then achieved by means of an arithmetic coding system that compresses binary symbols relative to an adaptive probability model. Once the entire image has been compressed, a post-processing operation determines the extent to which each code block's embedded bit stream should be truncated to achieve a particular target bit rate.

JPEG2000 was created as a new coding system for varying types of still imagery, with different characteristics. Where as JPEG, worked best with natural scenes, JPEG2000 was intended to work well with natural scenes, as well as imagery with text embedded. It also provides an approach that can compress data in either a lossless or lossy product, allowing for more flexibility within the LRIT system.

VII. Rice Compression

The Rice algorithm being used in the LRIT system is a lossless compression technique, recommended by CCSDS. The compression technique is well suited to the packet telemetry system being employed by the LRIT system. Following compression, the data structure that results is variable length and can be packetized as specified by the CCSDS recommendation on packet telemetry (Reference 4). The packets generated can then be transmitted over the satellite link and are received by the user station and the data recovered with a high reliability.

To achieve compression, the Rice algorithm takes advantage of a set of variable length codes. Each code is nearly optimal for a particular geometrically distributed source. The Variable length codes used by the Rice algorithm, compress data by assigning shorter code words to symbols that are expected to occur with higher frequency. By using several different codes and transmitting the code identifier, the Rice 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, and the performance of the algorithm is independent of packet size. The Rice algorithm is made up of a preprocessor to disassociate data samples and subsequently map them into symbols suitable for the entropy coding stage. The input data to the preprocessor, x, is a sample block of n-bit samples:

$x = x1, x2, . . . xJ$.

The preprocessor transforms input data into blocks of preprocessed samples, d, where:

$d = d1, d2, . . . dJ$.

The Adaptive Encoder converts preprocessed samples, d, into an encoded bit sequence y.

The entropy-coding module (Figure 1) 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 Rice algorithm can adapt to changing source statistics. Although the 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 imagery and non-imagery. It was determined that the achievable compression ratio was a function of the parameter J, which was set to 8, 16, 32, and 64 samples/block. 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 statistics.

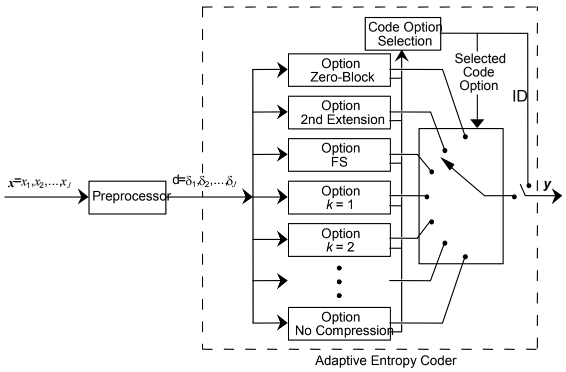

Figure 1

VIII. Zip Compression

ZIP compression is a variation of the Lempel-Ziv-Welch (LZW) compression technique. Like LZW, Zip scans a file or image to detect patterns in it, and replaces these by a single token or code. It is a lossless 'dictionary based' compression algorithm, that is, the algorithm scans a file for sequences of data that occur more than once. These sequences are then stored in a dictionary and within the compressed file; references are put wherever repetitive data occurred. Zip does not perform analysis of the incoming text; rather it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is output instead of a string of characters.

The code that is output can be of any arbitrary length, but it must have more bits in it than a single character. The first 256 codes (when using eight bit characters) are by default assigned to the standard character set. The remaining codes are assigned to strings as the algorithm proceeds.

Like LZW compression, Zip works best for files containing lots of repetitive data. This is often the case with text and monochrome images. Files that are compressed but which do not contain any repetitive information at all can sometimes grow larger.

IX. Conclusion

The digital LRIT service provides NOAA a level of flexibility and adaptability that it does not currently have with WEFAX, as such, the nature of digital data allows NOAA LRIT to incorporate various approaches to compression. The use of existing and mission specific header records provides an opportunity to exploit compression that best suits the data being broadcast to the user community. A sense of flexibility is gained by having the ability to determine which approach to compression is most appropriate to the data being processed. If the transmitted product can afford a slight difference from the original, then a lossy compression scheme can be used to best maximize the limited bandwidth. If the product is deemed highest priority and cannot afford to lose any data, then the implementation of Rice compression, using packet telemetry can be used. If the data were text, then the use of Zip compression would be a good choice. All in all, the LRIT specification provides the opportunity to be flexible and adaptable through the use of mission specific headers. It also allows the opportunity to adapt to new compression techniques as they become available.

X. References

CGMS LRIT/HRIT Global Specifications, Doc. No. CGMS 03, 1999

Lossless Data Compression, CCSDS Recommendation for Space data System Standards, CCSDS 121.0-B-1, May 1997

ISO/IEC IS 10918-1 | ITU-T Recommendation T.81

Packet Telemetry, CCSDS Recommendation for Space Data System Standards, CCSDS 102.0-B-4, November 1995

J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression", IEEE Transactions on Information Theory, May 1977