

**IEEE International Geoscience and Remote Sensing  
Symposium 2000**

July 24-28

**A REAL-TIME HIGH PERFORMANCE DATA  
COMPRESSION TECHNIQUE FOR SPACE  
APPLICATIONS**

**Pen-Shu Yeh\*, Jack Venbrux<sup>+</sup>, Prakash Bhatia<sup>+</sup>, Warner H. Miller\***

\*NASA/Goddard Space Flight Center

+Microelectronics Research Center, University of New Mexico

Paper presenter: Pen-Shu Yeh



# **A REAL-TIME HIGH PERFORMANCE DATA COMPRESSION TECHNIQUE FOR SPACE APPLICATIONS**

**Pen-Shu Yeh\*, Jack Venbrux<sup>+</sup>, Prakash Bhatia<sup>+</sup>, Warner H. Miller\***

**\*NASA/Goddard Space Flight Center**

**+Microelectronics Research Center, University of New Mexico**

## **ABSTRACT**

A high performance lossy data compression technique is currently being developed for space science applications under the requirement of high-speed push-broom scanning. The technique is also error-resilient in that error propagation is contained within a few scan lines. The algorithm is based on block-transform combined with bit-plane encoding; this combination results in an embedded bit string with exactly the desirable compression rate. The lossy coder is described. The compression scheme performs well on a suite of test images typical of images from spacecraft instruments. Hardware implementations are in development; a functional chip set is expected by the end of 2001.

## **KEY WORDS**

Coding, lossy data compression, telemetry, space science applications, push-broom instruments.

## **INTRODUCTION**

The advances in sensors and detectors have culminated in a new era of scientific instruments for space applications. These new instruments combine unprecedented spatial and signal resolutions in addition to offering multi-spectral capabilities. The result is an increase in the volume of data that has to be collected, buffered, transported, and archived in the space-to-ground data system.

To alleviate the burden caused by the extra data volume, compression has been suggested and in some cases implemented on space platforms. However, scientists often object to any loss in the data and thus accept only lossless data compression. The amount of data reduction achievable with lossless data compression is usually limited by the inherent entropy measurement in the data, and for many remote sensing applications is limited to about two-to-one.

For other applications that require higher data reduction, as in quick-look or direct-broadcast sensor data, a lossy compression technique is needed. There exist various algorithms that provide good



reconstruction performance at high compression ratios; however, none has adequately addressed the implementation requirement arising from push-broom instruments. A scan line of data produced by the push-broom type of sensor often exceeds two thousand pixels, and there is a limited amount of onboard buffering and processing time available before the data has to be transported for direct broadcast. In the following paragraphs, the requirements for space application of lossy data compression are stated, and then a description of a technique that meets all the requirements, along with its performance on several test images, is provided.

## **REQUIREMENTS FOR SPACE APPLICATIONS**

Any implementation of lossy data compression on a space platform has to satisfy several requirements. These requirements were addressed by the Consultative Committee for Space Data Systems (CCSDS) Subpanel 1A Compression Working Group in 1998. The mandatory requirements are:

1. Process non-frame based (push-broom) as well as frame based input source data.
2. Offer adjustable data rate.
3. Work with large source quantization ranges up to sixteen bits-per-pixel (bpp).
4. Offer real-time processing at or greater than twenty Msamples/sec, and at less than one watt/Msamples/sec. The power consumption includes all buffering and support electronics.
5. Require minimum ground interaction during operation.
6. Allow packetization for error containment.

A coder that meets all the above requirements is described below.

## **DESCRIPTION OF THE CODER**

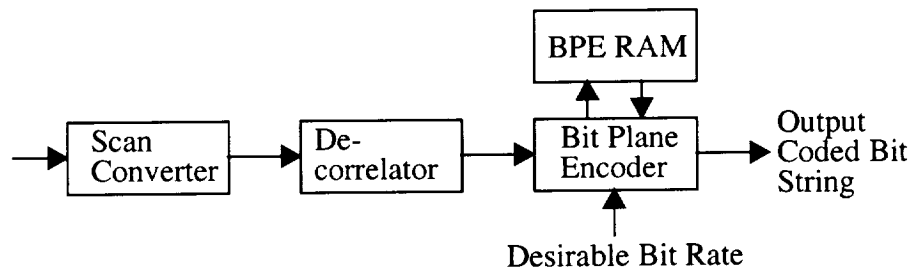
### **Overview**

The lossy coder consists of several functional modules depicted in Figure 1. The scan converter takes input imaging data and formats it into eight-by-eight blocks of integer values as input to the de-correlator. The de-correlator employs a hybrid transform that performs a size-8 modulated lapped transform (MLT) with sixteen input data points in the scan direction of imaging data, and a size-8 discrete cosine transform (DCT) in the other direction. This hybrid transform, termed enhanced DCT (EDCT), uses overlapping blocks in the scan line direction to reduce the blocking effect inherent in a two-dimensional (2D) DCT, but it allows isolation of strips of eight lines, as is often required by practical implementation in a packet data system. However, the system also allows an eight-by-eight 2D DCT, an eight-by-eight 2D MLT, or other types of block transforms to be implemented.

The bit plane encoder (BPE) first groups the eight-by-eight transform domain components into three family trees; each has one parent, four children, and sixteen grandchildren. The magnitudes of components are scanned for any most significant bit (MSB) on the scanned bit plane. This bit-plane scanning proceeds from the top-most bit plane downward. The positional information of those identified components is represented by a family tree structure and may be further coded for efficiency.



This information along with associated sign information is shifted to the output bit string from higher bit planes to lower bit planes.



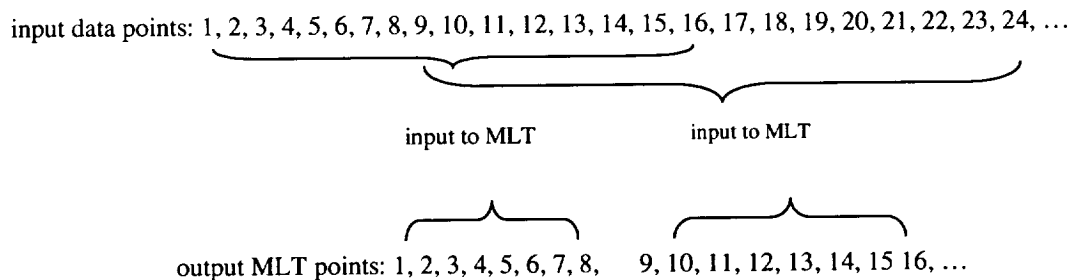
**Figure 1. Functional Diagram of the Coder**

The BPE random access memory (RAM) holds BPE-processed information for as many input blocks as it can support. The number of input blocks supported by this RAM is identified as one segment of input data. A segment can be as simple as one strip of eight lines, multiples of eight-line strips, or even half a strip.

The output bit string constitutes an embedded data format that allows progressive transmission and decoding to start at a lower bpp rate and proceed to a higher bpp rate. The bit string can be terminated at a desirable rate for precise control of output data rate.

### De-correlator

The EDCT is implemented by two one-dimensional transforms, the MLT and the DCT. The specified MLT takes sixteen input data points to provide eight transform components. The input is shifted by eight data samples each time a new MLT is performed, as shown in Figure 2.



**Figure 2. Input Output Data Points for MLT**





The MLT is performed with the following equation (see Reference 1):

$$X_m[j] = \sqrt{\frac{2}{8}} \sum_{i=0}^{15} x'_m[i] Z_m(i, j) \quad (1)$$

where  $m$  is used to index the MLT output block. The  $j$  index indicates the  $j$ th MLT component in block  $m$ . The input data sequence  $x[i]$  relates to the data samples in Equation (1) by:

$$x'_m[i] = x[(m - \frac{1}{2})8 + i] \quad (2)$$

The  $Z(i, j)$  function is given by:

$$Z_m(i, j) = -\sin[\frac{\pi}{16}(i + \frac{1}{2})] \cos[\frac{\pi}{8}(j + \frac{1}{2})(i + \frac{9}{2})] \quad 0 \leq i < 16 \quad (3)$$

for all blocks except the first and the last. For finite input data, the first and the last MLT blocks are implemented with the boundary conditions given in Reference 1.

The DCT is computed by:

$$X[j] = \sum_{i=0}^7 x[i] A(i, j) \quad (4)$$

$$A(i, j) = C_j \sqrt{\frac{2}{8}} \cos[\frac{j\pi}{8}(i + \frac{1}{2})] \quad (5)$$

with  $C_j$  having been defined earlier.

The inverse DCT is given by:

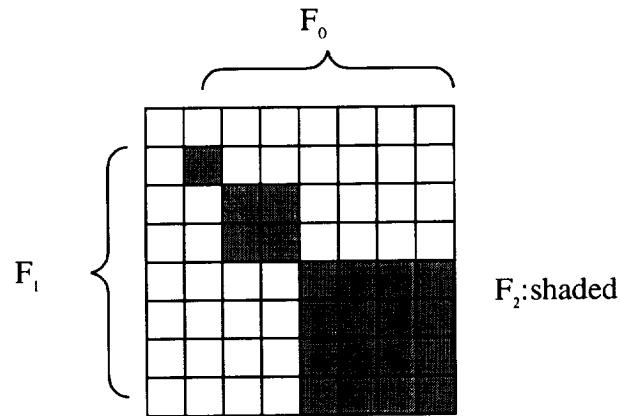
$$x[i] = \sum_{j=0}^7 X[j] A(i, j) \quad (6)$$

## Bit Plane Encoder

The frequency components of each block are then scanned from the highest bit plane of their binary representations. The purpose is to locate components of magnitude  $\geq 2^{nb}$  but  $< 2^{nb+1}$  at the  $nb$ -th bit plane (the lowest bit plane is when  $nb = 0$ ) and transmit their locations in the coded bit string before the information of components at lower bit planes is conveyed. In this coding scheme, quantization is inherently performed by an increase of a power of two as scanning proceeds from higher bit planes to



lower bit planes. To facilitate coding, a family tree structure at the  $nb$ -th bit plane, shown in Figure 3, is used to help identify components and guide the component scanning.



**Figure 3. Family Tree Structure**

### Family Tree Scanning

The tree structure in Figure 3 consists of three family members:  $F_0$ ,  $F_1$  and  $F_2$ . Each has one parent, four children, and sixteen grandchildren. During tree scanning, these elements are grouped into different types of patterns and each pattern scanned as one entity. Thus parents will be scanned first, followed by the children and the grandchildren.

For each block, the family tree scanning starts at the highest bit plane of all the AC components. At a particular bit plane,  $nb$ , information extracted from all blocks is concatenated before proceeding to a lower bit plane. This procedure produces an embedded bit stream that starts with information from higher bit planes and proceeds to information from lower bit planes. At the desirable compression bit rate, the bit stream can be stopped for precise rate control.

The performance of the BPE can be improved by further coding the scanned pattern from the family tree. A truncated lossless algorithm derived from the CCSDS Recommendation (Reference 2) is utilized in the coder.

## PERFORMANCE

The compression scheme was tested on different types of imaging data collected on space platforms. In the test, every strip of eight scan lines was processed independently and a fixed compression ratio was obtained for each strip of data. With the fixed compression ratio for each strip and without any information carried over strip boundary, error containment can be achieved within each strip.



Two de-correlators were used: the EDCT and the 2D DCT. The PSNR listed in Table 1 shows that EDCT provides a gain of 0.5 dB in most cases over the 2D DCT, except for the Hubble Space Telescope (HST) test images, where the results are reversed.

**Table 1. PSNR vs. Bit-Rate for Different Images**

Image	size	bits/ pixel	bpp 2.0	bpp 1.0	bpp 0.5	bpp 0.25	bpp 2.0	bpp 1.0	bpp 0.5	bpp 0.25
			EDCT				2D DCT			
forest(avhrr)	2048x2048	10	56.2	48.4	42.8	38.5	56.0	48.1	42.5	38.2
ice(avhrr)	2048x2048	10	52.9	46.3	41.8	38.2	52.5	45.8	41.3	37.9
india(avhrr)	2048x2048	10	50.5	43.0	37.9	34.1	50.0	42.5	37.5	33.7
ocean(avhrr)	2048x2048	10	50.6	43.8	39.2	35.5	50.3	43.5	38.8	35.1
solar	1024x1024	12	54.7	49.0	44.7	41.4	54.3	48.6	44.4	41.2
sunspot	512x512	12	59.1	54.6	50.9	45.9	58.8	54.2	50.2	45.1
wfpc(hst)	800x800	12	71.5	67.8	64.5	62.5	72.2	68.4	66.2	64.4
foc(hst)	1024x512	12	71.8	66.8	64.2	62.6	72.1	67.1	64.4	62.8
mars	512x512	8	41.0	34.5	29.9	26.3	40.2	33.8	29.4	25.9

The performance of the compression can be improved by using larger BPE RAM to buffer intermediate results from more than one strip; however, dependency among these strips will result. For comparison, a 2D MLT was applied to the "india" test image, and a frame buffer was allocated. The improvement in PSNR at 2.0 bpp is about 2.3 dB.

## TECHNOLOGY STATUS

Rad-hard implementation of the compression scheme is being pursued. Currently, a custom VLSI chip for performing EDCT and the 2D DCT has been designed, fabricated. Its processing speed is expected to exceed twenty-five Msamples/sec. The VLSI architecture for BPE is under study. A functional chip is expected in 2001. The compression scheme is simulated in software and has been tested on various types of space data with satisfactory results.

## CONCLUSION

A high performance lossy data compression scheme has been developed for space applications. This scheme offers real-time (over twenty Msamples/sec) processing on push-broom types of instruments. The technique produces an embedded bit string with the desirable features of precise rate control and no operator intervention.



## REFERENCES

- 1 Henrique S. Malvar. *Signal Processing with Lapped Transform*. Norwood, Massachusetts: Artech House, 1992.
- 2 *Lossless Data Compression*. Recommendation for Space Data System Standards, CCSDS 121.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1997.

