Compression algorithm and implementation for the PRISMA mission

COMPRESSION ALGORITHM AND IMPLEMENTATION FOR THE PRISMA MISSION

Diego Valsesia[1], **Maurizio De Nino**[2], **and Enrico Magli**[3]

[1]*Politecnico di Torino*
[2]*Techno System Dev. s.r.l.*
[3]*Politecnico di Torino*

ABSTRACT

In this paper we describe the image compression algorithm and its implementation to be used for the PRISMA mission of the Italian Space Agency. The mission payload includes a pushbroom hyperspectral instrument as well as a medium resolution panchromatic camera.

1. INTRODUCTION

PRISMA (PRecursore IperSpettrale della Missione Applicativa) is a mission of the Italian Space Agency to serve as a pre-operational technology demonstrator for hyperspectral payloads in space. The mission payload includes a pushbroom hyperspectral instrument with 237 bands as well as a medium resolution panchromatic camera. The mission specifications require a compression ratio of 1.5:1 to match the downlink capacity, therefore requiring a compression board to be included in the payload. Moreover, the compression board must satisfy the target throughput of 60 Msamples/s. This will be obtained employing an algorithm based on a prediction loop with quantization and entropy coding. For this compressor, the algorithm will be an extension of the recently published CCSDS-123 lossless compression recommendation [1]. Despite having been standardized very recently, this recommendation is based on an algorithm developed by NASA-JPL [2]. NASA has already demonstrated a hardware implementation of this algorithm. The requirement on compression ratio is such that lossless compression is typically not enough, so a lossy operating mode must be provided. In a recent ESA project, the CCSDS-123 recommendation has been extended to handle lossy compression and rate control [3] and the resulting compression algorithm has been implemented on a space-qualified FPGA. Its validation, including a complete image acquisition and processing pipeline with frame grabber and CCSDS formatter, with the compression algorithm implemented on Xilinx 5QV FX130T FPGA, has achieved a throughput of 20 Msamples/s [4]. In this paper, we address how the the previous work on a lossy extension of CCSDS-123 can be tailored to meet the requirements of the PRISMA mission in terms of compression ratio and, even more critically, throughput. This is accomplished by using a modified version of a prediction mode specified by CCSDS-123 that allows a faster hardware implementation. Moreover, we study which entropy coder among a modified range encoder working in parallel on bitplanes and a Golomb encoder is more suitable for the mission. As a further addition to the original algorithm, PRISMA requires resilience to transmission errors. This is integrated into the prediction loop with a reset of the prediction parameters and of the entropy encoding in order to create independent coding units.

2. PRISMA SENSORS

PRISMA comprises two spectrometers are used to acquire images in the visible and near-infrared range (VNIR) and short-wave infrared range (SWIR) as well as a panchromatic camera. All the sensors produce registered outputs. The VNIR and SWIR spectrometers have a spatial resolution of 30 m on a swath of 30 km and a spectral resolution of 12 nm, while the panchromatic camera has a spatial resolution of 5 m. The VNIR spectrometer is sensitive to the wavelengths in the $[400, 1010]$ nm range, subdivided into 66 bands. The VNIR spectrometer is instead sensitive to the wavelengths in the $[920, 2505]$ nm range, subdivided into 171 bands.

3. REVIEW OF CCSDS-123

The Consultative Committee for Space Data Systems (CCSDS) has recently developed the CCSDS-123 recommendation, intended for lossless compression of multispectral and hyperspectral images. CCSDS-123 is based on the Fast Lossless compression algorithm [5] [2], which is a predictive method. The algorithm computes a local sum $\sigma_{z,y,x}$, obtained from a causal neighborhood of the pixel. A weighted combination of the local sums in the P previous bands yields the predicted pixel value. The algorithm adapts the weights using the sign algorithm [6], which is a low-complexity solution for the implementation of a least-mean-square filter.

Let $s_{z,y,x}$ denote the pixel value at position (x, y, z), then

the encoder computes:

$$\hat{d}_{z,y,x} = \mathbf{W}_{z,y,x}^T \mathbf{U}_{z,y,x}$$

Under reduced prediction mode:

$$\mathbf{U}_{z,y,x} = \begin{bmatrix} d_{z-1,y,x} \\ d_{z-2,y,x} \\ \vdots \\ d_{z-P,y,x} \end{bmatrix},$$

while under full prediction mode directional differences are also used:

$$\mathbf{U}_{z,y,x} = \begin{bmatrix} d^N_{z,y,x} \\ d^W_{z,y,x} \\ d^{NW}_{z,y,x} \\ d_{z-1,y,x} \\ d_{z-2,y,x} \\ \vdots \\ d_{z-P,y,x} \end{bmatrix}.$$

The differences are defined as $d_{z,y,x} = 4s_{z,y,x} - \sigma_{z,y,x}$, while the directional differences are:

$$d^N_{z,y,x} = \begin{cases} 4s_{z,y-1,x} - \sigma_{z,y,x}, & y > 0 \\ 0, & y = 0 \end{cases}$$

$$d^W_{z,y,x} = \begin{cases} 4s_{z,y,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & y = 0 \end{cases}$$

$$d^{NW}_{z,y,x} = \begin{cases} 4s_{z,y-1,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & y = 0 \end{cases}$$

The local sums $\sigma_{z,y,x}$ are defined differently depending on the choice between neighbor-oriented sums and column-oriented sums. Under column-oriented mode:

$$\sigma_{z,y,x} = \begin{cases} 4s_{z,y-1,x}, & y > 0 \\ 4s_{z,y,x-1}, & y = 0, x > 0 \end{cases}$$

while under neighbor-oriented mode:

$$\sigma_{z,y,x} = \begin{cases} s_{z,y,x-1} + s_{z,y-1,x-1} + s_{z,y-1,x} + s_{z,y-1,x+1} \\ \qquad y > 0, 0 < x < N_x - 1 \\ 4s_{z,y,x-1}, \quad y = 0, x > 0 \\ 2s_{z,y-1,x} + 2s_{z,y-1,x+1}, \\ \qquad y > 0, x = 0 \\ s_{z,y,x-1} + s_{z,y-1,x-1} + 2s_{z,y-1,x}, \\ \qquad y > 0, x = N_x + 1 \end{cases}$$

A *scaled predicted sample* $\tilde{s}_{z,y,x}$ is calculated from $\hat{d}_{z,y,x}$. The prediction residual is computed as $\Delta_{z,y,x} = s_{z,y,x} - \left\lfloor \frac{\tilde{s}_{z,y,x}}{2} \right\rfloor$ and then mapped to a positive integer $\delta_{z,y,x}$ to be entropy encoded.

The entropy coding stage provides two options: a block-adaptive method, which is intended for backward compatibility with the CCSDS 121.0B standard, a sample-adaptive method. The sample-adaptive entropy coder maintains separate entropy coding statistics for each spectral band, thus producing the same compressed image size regardless of the order in which samples are presented to the encoder. It is typically more efficient than the block-adaptive coder. Each mapped prediction residual is encoded into a variable length codeword according to the Golomb coding. Essentially, a non-negative value δ is divided by a tunable parameter k to obtain the result of the division q and a remainder r . Then,

- q is written using unary coding, i.e. a sequence of $q - 1$ zeros terminated by a 1
- r is written using truncated binary coding

In the CCSDS 123.0-B-1 implementation, the parameter k is adaptive and it is computed from statistics on the mapped residuals kept individually for each band. In particular, an accumulator and a counter are defined. The initial counter value for all bands is defined as $\Gamma(0) = 2^{\gamma_0}$, where γ_0 is a user-defined value in the range $1 \leq \gamma_0 \leq 8$. The initial accumulator value for band z is defined as:

$$\Sigma_z(0) = \lfloor 2^{-7} \left(3 \cdot 2^{k+6} - 49\right) \Gamma(0) \rfloor$$

where k is the user-defined initialization value of the aforementioned parameter, in the range $0 \leq k \leq D2$, being D the dynamic range. After coding a mapped residual, the counter is updated as follows:

$$\Gamma(t) = \begin{cases} \Gamma(t-1) + 1, & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \lfloor \frac{\Gamma(t-1)+1}{2} \rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases}$$

where γ^* is a user-defined parameter in the range $4 \leq \gamma^* \leq 9$. After coding a mapped residual δ , the accumulator is updated as follows:

$$\Sigma_z(t) = \begin{cases} \Sigma_z(t-1) + \delta, & \Gamma(t-1) < 2^{\gamma^*} - 1 \\ \lfloor \frac{\Sigma_z(t-1)+\delta+1}{2} \rfloor, & \Gamma(t-1) = 2^{\gamma^*} - 1 \end{cases}$$

The value of the divisor for the current sample is the largest nonnegative integer $k_z(t) \leq D2$ such that $\Gamma(t)2^{k_z(t)} \leq \Sigma_z(t) + \lfloor \frac{49}{2^7}\Gamma(t) \rfloor$ The codeword will have $u_z(t)$ zeros followed by a one, where $u_z(t) = \lfloor \frac{\delta}{2^{k_z(t)}} \rfloor$. Finally, the rest of the codeword is composed of the $k_z(t)$ least significant bits of δ. An exception to this procedure is made when $u_z(t) \geq u_{max}$, being u_{max} a user-defined constant. In such case u_{max} zeros are written and followed by the D-bit binary representation of δ.

For further details, we refer the reader to the CCSDS-123 Blue Book [1] and to the paper by Augé *et al.* [7] for a more throughout explanation of the encoder parameters and their impact on performance.

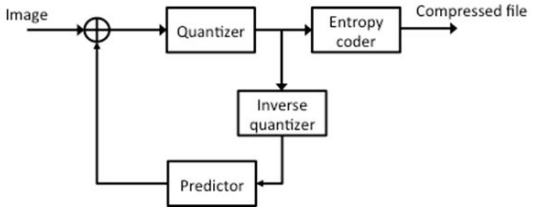

Figure 1. Prediction loop with quantization

4. PROPOSED METHOD

The compressor in [3] can be described using the block diagram in Fig. 1. The algorithm comprises a spatial-spectral predictor, a quantizer and an entropy coding stage. The quantizer is inserted in a quantization feedback loop. A local decoder generates reconstructed samples, which are used to predict the next input sample. The prediction residual is then quantized, and the quantized value is compressed into a codeword by the entropy coder, and written to file. Quantization step sizes employed during the compression stage are either implicitly known by the decoder (e.g., recalculated from past decoded samples), or explicitly written in the compressed file. If the quantizer is bypassed, the compressor works in purely lossless mode and the reconstructed image is identical to the original. Increasing quantization step sizes will lead to increasing information losses, but also larger amounts of compression.

4.1. Predictor

The CCSDS123 recommendation defines two prediction modes: "full" and "reduced", with two modes to compute local sums, i.e. column-oriented and neighbor-oriented.

The prediction mode used by PRISMA is the reduced, column-oriented predictor defined in CCSDS 123.0-B-1 as this is the choice that allows implementations with highest throughput. The initialization procedure described by CCSDS 123.0-B-1 has been modified in order to avoid the use of the sample immediately to the left of the current sample in the current band. Indeed, while guaranteeing excellent compression performance, this choice avoids employing for the prediction any pixel located on the same line as the current pixel. This enables pipelining the compressor operation in a much more efficient way, allowing to achieve the desired throughput for the prediction stage. Moreover, the prediction weights are reset to their initial values after encoding a predetermined number of lines N. This solution embeds resilience to packet losses in the compressed stream. The initialization of the predictor must be repeated after every reset.

Following the same notation of the recommendation [1], the local sum is defined in the following way:

$$\sigma_{z,y,x} = \begin{cases} 4s_{z,y-1,x} & \text{for } y \bmod N > 0 \\ 4s_{z-1,y,x-1} & \text{for } y \bmod N > 0, z > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Note that $\sigma_{z,kN,0}$ is undefined for $z > 0$ and $k = 0, 1, 2, \ldots$. This is handled by the modified computation of the scaled predicted sample. The scaled predicted sample value is initialized in the following way:

$$\tilde{s}_{z,y,x} = \begin{cases} 2s_{mid} & \text{for } x \geq 0, z = 0, y \bmod N = 0 \\ 2s_{0,y,z-1} & \text{for } x = 0, z > 0, y \bmod N = 0 \end{cases} \quad (2)$$

where $s_{mid} = 2^{D-1}$, being D the dynamic range.

In order to improve resilience to packet losses during data transmission, the prediction loop is broken after a predetermined number of lines. This allows to partition the image into independent pieces, so that is a packet is lost only a portion of the image is corrupted. The reset mechanism requires the following operations to be performed whenever $y \equiv 0(\text{mod}N)$, i.e. at the beginning of the next line after N lines have been coded :

- reset the predictor weights to their default value
- reset the statistical models of the entropy encoder
- pad the output to reach an integer number of bytes
- write a synchronization marker to the output file

4.2. Quantizer

Quantization in the PRISMA compressor architecture will be a simple scalar uniform quantization, with the same quantization step size employed for every pixel in every band of the image. This choice corresponds to the so-called nearl-ossless compression, which has the desirable feature to provide a strict upper bound on the maximum absolute error incurred between any pixel of the original image and the corresponding pixel in the reconstructed image. It is foreseen that only small quantization step sizes will be employed, corresponding to very small maximum errors. The exact values of these quantization step sizes will be chosen based on the characteristics of the images to be generated by PRISMA. Moreover, the choice of a constant quantization step size for every pixel avoids the need for a rate control algorithm. This will ensure that quantization does not act as a bottleneck on the compressor throughput.

4.3. Entropy coder

The entropy coding stage will be designed so as to achieve excellent compression performance, while

achieving a very high throughput. HYDRA [3] has been the first encoder validated for space that employs a range encoder, i.e. a simplified version of the arithmetic coder, which obtains close to optimal compression efficiency. Indeed, HYDRA employed four parallel range encoders to improve its throughput. With respect to HYDRA, the following options for entropy coding have been considered in order to further increase the throughput.

- To employ 16 parallel binary range encoders instead of four nonbinary ones. A binary range encoder is faster and occupies fewer hardware resources. Each binary range encoder shall be applied to a bitplane, i.e. the set of all bits of equal weight in the binary representation of the prediction residuals of an image. Since range encoders need to adaptively estimate the statistics of prediction residuals, it is expected that 16 binary encoders (requiring to estimate 32 probabilities overall) will work as well as or better than nonbinary encoders, whose transient for probability adaptation takes a longer time.
- Another viable option, which will certainly achieve very high throughputs, is to use a Golomb power -of-two code coder. The Golomb coder has been a classical choice for onboard image compression, because of its simplicity leading to high throughput, coupled with very good performance. The main issue with the Golomb coder lies in the fact that it cannot generate bitrates smaller than 1 bit/pixel. However, the compression ratios of interest for PRISMA are above 6 bit/pixel; at these bitrates the Golomb coder will work very well, providing very good coding efficiency.

During implementation it emerged that the solution using 16 binary range coders in parallel could not meet the required 60 Msamples/s target. Hence, the entropy coding stage adopts the sample adaptive coder based on Golomb-Power-of-two codes as defined in the CCSDS 123.0-B-1 standard.

5. EXPERIMENTAL RESULTS

The following results have been obtained using images generated from the sensor simulation. Only the bands with nonzero content have been supplied to the compression algorithm and all the rates in bits per pixel refer to the actual number of nonzero bands. The images are split in their short-wave infrared (SWIR) and visible and near infrared (VNIR) as they will be coded separately by parallel instances of the compression hardware. The goal of the experiments is to verify that the target compression ratio can be met by the proposed architecture and to measure the quality reached on the test images. Moreover, we quantify the impact of prediction reset in terms of output rate for different choices of the constant N. The quality metrics that we use are Signal-to-Noise ratio (SNR), defined as:

$$\mathrm{SNR} = 10 \log_{10} \frac{\sum_{i=1}^{N_{pixels}} x_i^2}{\sum_{i=1}^{N_{pixels}} (x_i - \hat{x}_i)^2},$$

as well as the Maximum Absolute Distortion (MAD), which is the maximum magnitude of the error on a pixel. Since we are using near-lossless compression, the MAD must be bounded by half the quantization step size.

5.1. Results on clean images

We first test the algorithm on clean images, i.e. images presenting no sensor defects such as bright or dark pixels. The maximum acceptable rate to satisfy the compression ratio constraint is about 8 bpp. Table 1 reports the results on some of the images for different quantization step sizes. The quantization step size is obtained from the parameter δ as $Q = 2\delta + 1$. It can be noticed that the rate constraint is always met even by lossless compression and that the reset mechanism introduces a small overhead.

5.2. Results on images with defective pixels

The following results consider images where some pixels in the detector are defective. Hence, a column of the image has been zeroed, and its position changes with different bands. The goal of this experiment is to verify that the compression algorithm behaves correctly even in presence of artifacts in the image. Table 2 shows that the rate constraint is satisfied in this case as well with a minimal increase in rate from Table 1.

Rate constraint for each line

In addition to the global rate requirement of 8 bpp (1.5:1 compression ratio), the hardware architecture constrains the rate of each line with all the spectral channels to be below 9.2 bpp (1.3:1 compression ratio). This requirement may pose challenges when the prediction is initialized, such as for the first line and very time it is reset. Fig. 5.2 shows the behaviour of the output rate for each line for some of the test images for lossless compression. It can be noticed that despite the peaks due to the reset of the predictor every 16 lines, the output rate never reaches the critical value of 9.2 bpp.

REFERENCES

1. Consultative Committee for Space Data Systems (CCSDS), "Lossless Multispectral and Hyperspectral Image Compression," *Blue Book*, no. 1, May 2012.

Table 1. Results on clean images

		NO RESET			RESET 16			RESET 64		
Image	DELTA	RATE	SNR (dB)	MAD	RATE	SNR (dB)	MAD	RATE	SNR (dB)	MAD
SWIR_Cuprite	0	6.769	inf	0	6.951	inf	0	6.822	inf	0
	1	5.343	73.31	1	5.470	73.31	1	5.360	73.31	1
	2	4.650	68.54	2	4.789	68.53	2	4.674	68.53	2
	3	4.187	65.51	3	4.331	65.52	3	4.219	65.52	3
SWIR_ErtaAle	0	6.876	inf	0	7.067	inf	0	6.933	inf	0
	1	5.452	74.80	1	5.575	74.80	1	5.455	74.80	1
	2	4.722	70.02	2	4.888	70.03	2	4.758	70.03	2
	3	4.289	67.02	3	4.431	67.01	3	4.312	67.01	3
VNIR_Cuprite	0	4.218	inf	0	4.351	inf	0	4.254	inf	0
	1	3.295	54.26	1	3.235	54.26	1	3.131	54.26	1
	2	2.591	49.48	2	2.660	49.48	2	2.524	49.48	2
	3	2.391	46.45	3	2.318	46.44	3	2.220	46.45	3
VNIR_ErtaAle	0	4.449	inf	0	4.586	inf	0	4.475	inf	0
	1	3.412	57.66	1	3.392	57.66	1	3.263	57.66	1
	2	2.748	52.89	2	2.803	52.89	2	2.658	52.89	2
	3	2.471	49.87	3	2.420	49.87	3	2.298	49.87	3

Table 2. Results on images with defective pixels

		NO RESET			RESET 16			RESET 64		
Image	DELTA	RATE	SNR (dB)	MAD	RATE	SNR (dB)	MAD	RATE	SNR (dB)	MAD
SWIR_Cuprite	0	6.770	inf	0	6.958	inf	0	6.824	inf	0
	1	5.546	73.30	1	5.667	73.30	1	5.564	73.30	1
	2	4.887	68.53	2	5.003	68.53	2	4.898	68.53	2
	3	4.428	65.51	3	4.551	65.51	3	4.451	65.51	3
SWIR_ErtaAle	0	6.876	inf	0	7.073	inf	0	6.934	inf	0
	1	5.620	74.80	1	5.755	74.80	1	5.641	74.80	1
	2	4.945	70.03	2	5.088	70.03	2	4.969	70.03	2
	3	4.496	67.02	3	4.633	67.01	3	4.522	67.01	3
VNIR_Cuprite	0	4.218	inf	0	4.351	inf	0	4.254	inf	0
	1	3.295	54.26	1	3.235	54.26	1	3.131	54.26	1
	2	2.591	49.48	2	2.660	49.48	2	2.524	49.48	2
	3	2.391	46.45	3	2.318	46.44	3	2.220	46.45	3
VNIR_ErtaAle	0	4.449	inf	0	4.586	inf	0	4.475	inf	0
	1	3.412	57.66	1	3.392	57.66	1	3.263	57.66	1
	2	2.748	52.89	2	2.803	52.89	2	2.658	52.89	2
	3	2.471	49.87	3	2.420	49.87	3	2.298	49.87	3

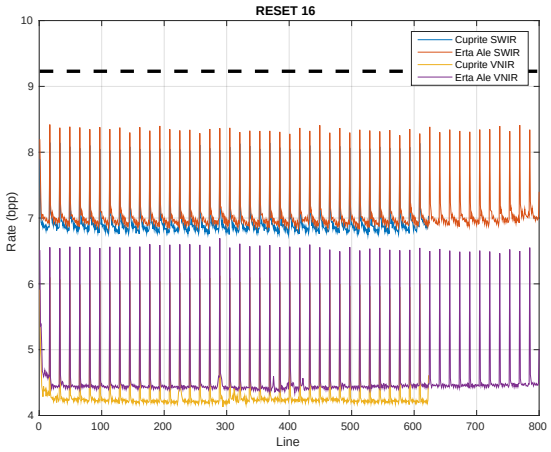

Figure 2. Output rate per line with prediction reset

2. A. B. Kiely and M. A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2672–2678, 2009.

3. D. Valsesia and E. Magli, "A novel rate control algorithm for onboard predictive coding of multispectral and hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 6341–6355, Oct 2014.

4. M. De Nino, G. Capuano, M. Romano, and E. Magli, "Lossy multi/hyperspectral compression hw implementation at high data rate," in *Proc. of International Astronautical Congress*, 2014.

5. M. A. Klimesh, "Low-complexity lossless compression of hyperspectral imagery via adaptive filtering," 2005.

6. S. H. Cho and V. J. Mathews, "Tracking analysis of the sign algorithm in nonstationary environments," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 12, pp. 2046–2057, 1990.

7. E. Augé, J. Sánchez, A. Kiely, I. Blanes, and J. Serra-Sagristà, "Performance impact of parameter tuning on the CCSDS-123 lossles multi- and hyperspectral image compression standard," 2013.