An Hardware Implementation of a Novel Algorithm For Onboard Compression of Multispectral and Hyperspectral Images

4th International Workshop
on
On-Board Payload Data Compression

23 & 24 October 2014
Venice, Italy

Maurizio De Nino[(1)], Giuseppe Capuano[(1)], Mario Romano[(1)], Enrico Magli[(2)]

[(1)] Techno System Developments
via Provinciale Pianura 2 int. 23 - 80078 Pozzuoli (Naples), Italy
Email:
mdenino@tsd-space.it
gcapuano@tsd-space.it
mromano@tsd-space.it

[(2)]Politecnico di Torino
corso Duca degli Abruzzi, 24 - 10129 Torino, Italy
Email: enrico.magli@polito.it

INTRODUCTION

New multispectral and hyperspectral instruments are going to generate very high data rates due to the increased spatial and spectral resolution. In this context, the compression is a very important part of any onboard data processing system for Earth observation and astronomical missions.
More recently, lossless compression has started to be routinely used for spaceborne Earth observation satellites. The CCSDS has established a working group (WG) on Multispectral and Hyperspectral Data Compression (MHDC), which has the purpose of standardizing compression techniques to be used onboard. The WG has already standardized a lossless compression algorithm for multispectral and hyperspectral images, and has started working on a lossy compression algorithm.
Under an ESA contract, aimed to investigate new techniques for Lossy multi/hyperspectral compression for very high data rate instruments (HYDRA), TSD in collaboration with Politecnico of Torino, designed an IP core for FPGA and/or ASIC implementation of a lossy compression algorithm. In addition to the IP core, TSD developed a HW platform based on the Xilinx Virtex-5 XQR5VFX130, the industry's first high performance rad-hard reconfigurable FPGA for processing-intensive for space systems. Advanced results along with details of electronic platform design will be presented in this paper.

COMPRESSION ALGORITHM

The implemented multispectral and hyperspectral images compression algorithm is based on the predictive lossy compression paradigm. Predictive lossy compression can be seen as derived from lossless ones, by simply applying a uniform quantizer to the prediction residuals.
The predictive lossy compression is able to achieve compression performance as good as the transform-based approach, at a fraction of its complexity and providing a very good fit to onboard image compression. Although predictive schemes have several potential advantages over transforms, it should be noted that they cannot achieve rate control (i.e., coding at a predetermined bit-rate) in a way as simple as can be done using transforms. This does not mean that rate control with predictive coders is not possible, but it is probably not easy or accurate as would be if a transform were used. The other problem is that predictive lossy compression requires an entropy coder such as an arithmetic coder, to achieve low bit-rates; the arithmetic coders are rather complex from the computational standpoint.
The implemented algorithm can be considered as an extension of CCSDS 123 [8] with a quantization stage to achieve lossy compression and a modified entropy coder (range encoder) (See Figure 1).

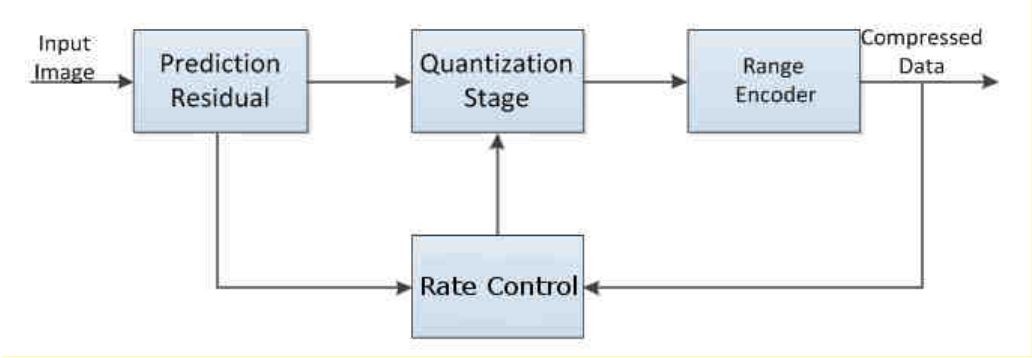

Figure 1: Logic block diagram of compressor

FLIGHT HW PLATFORM

The test environment is based on a flight hardware to validate not only the algorithm but also its onboard implementation. The flight hardware is the HPHC (High performance Processing unit for Hyperspecral data Compression), belonging to a family of High performance Processing unit (HPxx) developed by TSD for real time image data processing (See Figure 2).

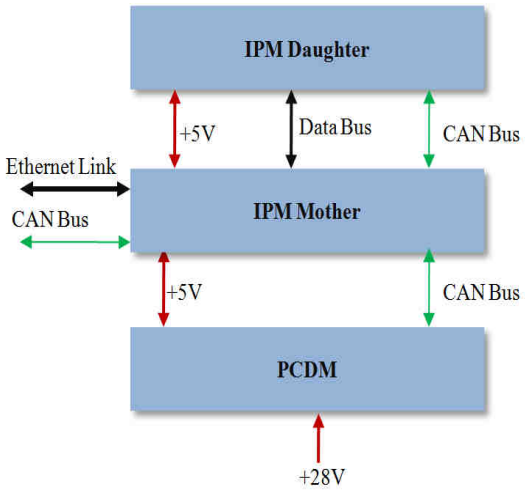

Figure 2: Basic block diagram of HPHC

The platform includes the following modules:

- Image Processing Module (IPM)
- Power Conditioning & Distribution Module (PCDM)

The IPM is composed of four sections as shown in figure 3:

- Image Processing A
- Image Processing B
- Data Handling A
- Data Handling B

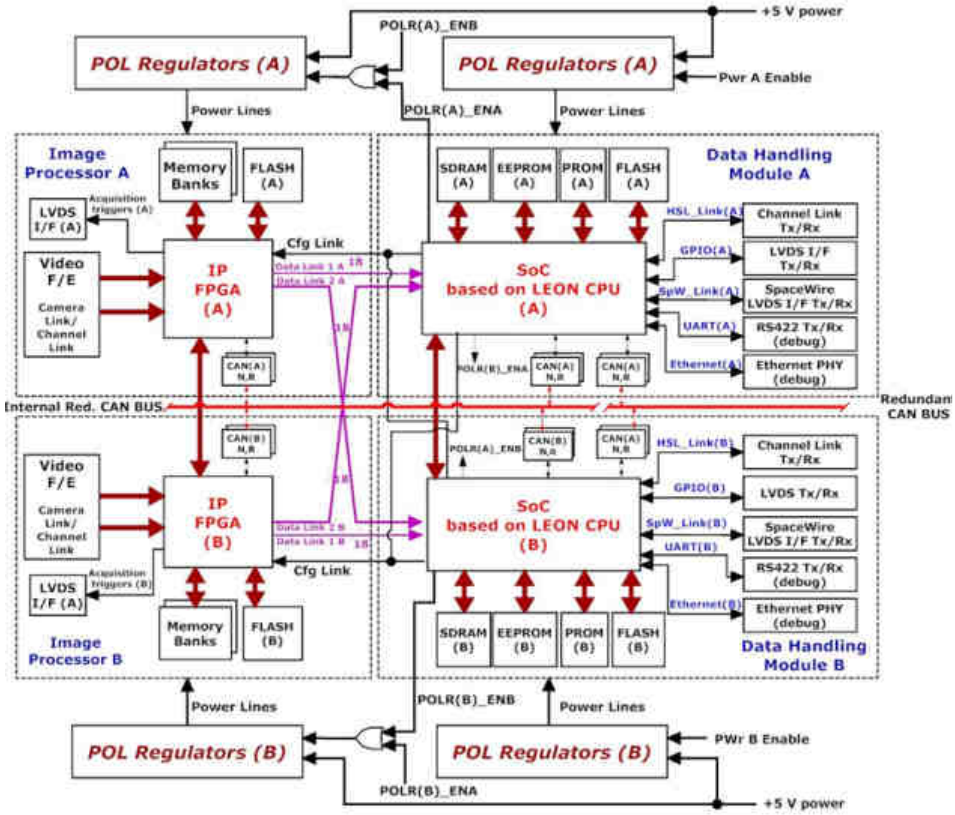

Figure 3: Detailed scheme of IPM

The two Image Processing sections and the two Data Handling sections are identical and they can be configured:

- in cold redundancy to provide high reliability
- in master-slave mode to run in parallel so to improve the processing capabilities

The Data Handling section implements the control & communication functionalities of the unit, while the Image Processing section is dedicated to the compression.
As shown in Figure 3, the outputs of the Image Processor sections are redounded and connected in cross-strapped mode to both Data Handling sections. Point-to point links are also available to interconnect the Image Processing A section with the Image Processing B and the Data Handling A with the Data Handling B, thus allowing the sections to run in parallel.

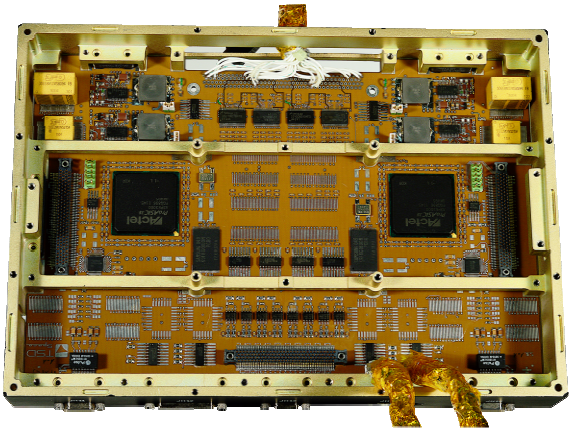

Figure 4: IPM Mother board. Note on both sides the anti-fuse FPGAs

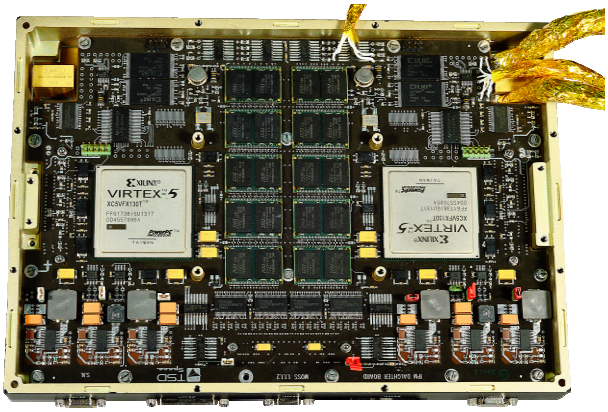

Figure 5: IPM Daughter board. On both sides can be seen the two Virtex5 FPGAs

Each section of image processing adopts the Xilinx Virtex-5 XC5VFX130T (XQR5VFX130 for flight Model), the industry's first high performance rad-hard reconfigurable FPGA.
Each FPGA is provided with 5Gbit SDRAM and two image data inputs (1.575 Gbits/s each).

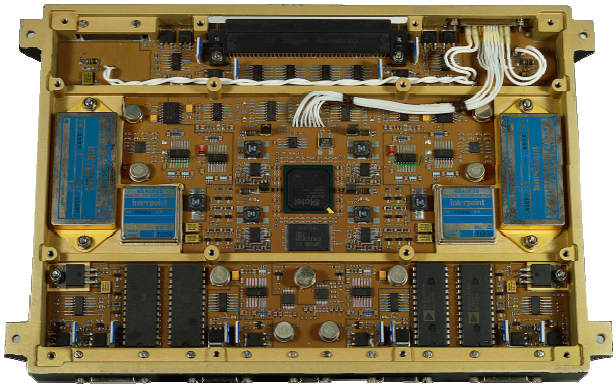

Figure 6: PCDM

Figure 7: HPHC engineering model assembled in its cabinet

HYDRA IP CORE

The core of Multispectral and Hyperspectral Data Compressor is implemented only on Processor Image Unit A since a single Virtex5 chip has enough hardware resources that are more than sufficient for our purposes.
Figure 8 describes the basic block diagram of compressor.

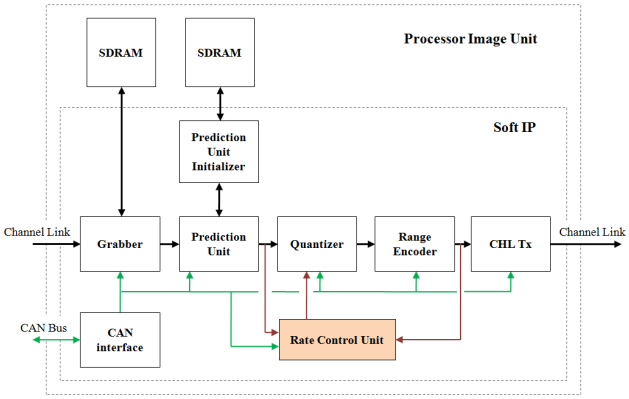

Figure 8: Basic block diagram of compressor

Once a Cube Of Images (COI) is received, it is processed by the Grabber that transmits a pair of pixels at a time to the Prediction Unit (PU). The two pixels belong to two adjacent rows at the same column. For this reason, considering the maximum size of 4096 columns x 4096 bands @16bits, the maximum amount of memory necessary for grabbing is 32Mbytes since it is necessary to store the previous horizontal plane of the COI.
Figure 9 shows a generic COI; the current horizontal plane is received pixel by pixel and at same time the grabber stores these pixels on SDRAM and provides the pixels of previous horizontal plane by reading them from SDRAM. At the end of plane the cycle continues to the next plane until the end of COI. When the first plane is transmitted, the previous plane doesn't exists yet, therefore the pixel of the previous plane of the pair to transmit has its value forced to 0x0000.

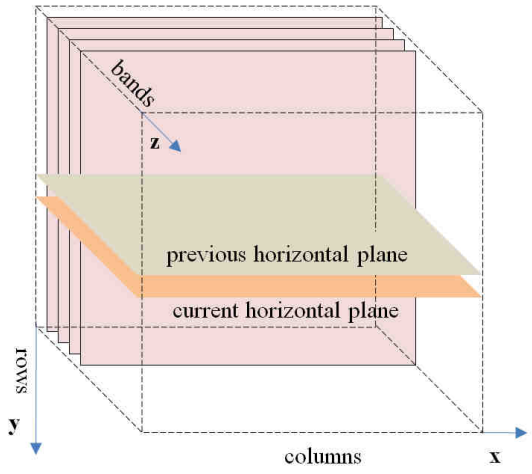

Figure 9: Anatomy of a COI according to BIL format

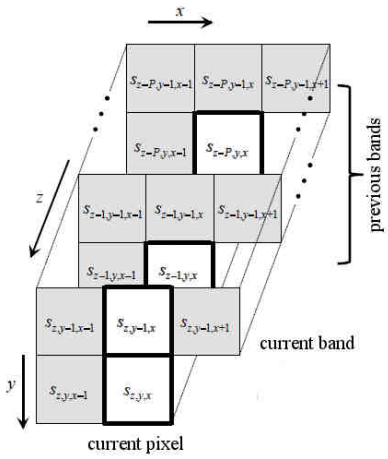

Figure 10: Pixels taken in account to calculate the predicted value of the current pixel

The PU calculates, for each pair of pixels in input, the predicted value of pixel belonging to the current horizontal plane. The predicted value is subtracted to the real value and this residual value, after the quantization, thanks to its lower information content, can be compressed with higher efficiency.
Figure 10 shows how the predicted value is calculated [2]. We tell $S_{z,y,x}$ the current pixel, the related predicted value is calculated using the values of pixels of the five previous bands having the same x,y coordinates ($S_{z-1,y,x}$, $S_{z-2,y,x}$..... $S_{z-5,y,x}$) and the pixel of previous row $S_{z,y-1,x}$.
The PU makes use of a SDRAM to store the weights vectors that are updated continuously. Every time a new COI has to be processed, the starting values of weights vectors are written into SDRAM by means of the Prediction Unit Initializator.
The stage after the prediction is the quantization. When the quantization is enabled, the compression mode is lossy. The compressor can operate in loss-less mode by disabling the Quantizer.

All the residuals (with or without quantization) are encoded by the Range Encoder, an entropy encoder that remove the numerical redundancies making use of statistics. Figure 11 illustrates the simplified block diagram of the range encoder. Four different independent statistic models (SM) are used; each SM stores and updates the following statistical data:

- The cumulative frequency for each symbol
- The frequency for each symbol
- Total frequency defined as the sum of overall frequency of symbols

Different data structures can be used to organize cumulative frequencies, each one with its own advantages and drawbacks. In our implementation it was adopted the Cumulative Frequency Matrix (CFM) [1] which requires a limited HW resources and it is very fast. CFM is implemented by using a built-in RAM on FPGA.
The total frequency of each SM is managed making use of a counter. The encoding process consist of following phases:

- identification of the symbols to encode starting from the current residual value
- extraction of statistical data for each identified symbol
- processing of statistical by using a complex arithmetic circuit
- update the Statistical Models
- rescaling the values of cumulative and total frequencies. The rescaling is necessary to avoid overflows of SMs elements.

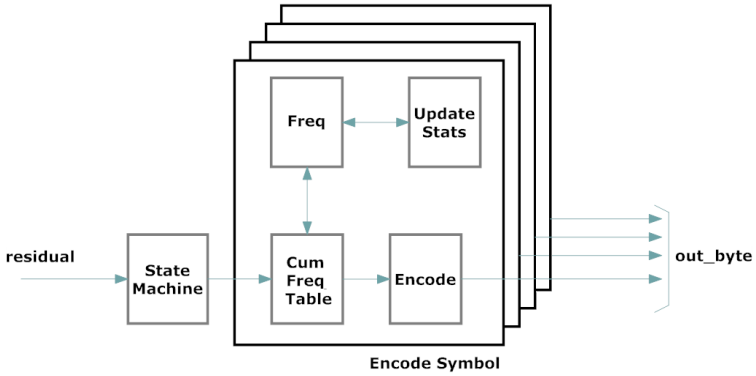

Figure 11: Basic block diagram of Range Encoder

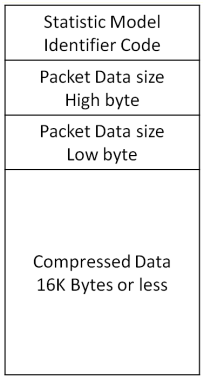

Figure 12. Composition of single data packet

The biggest challenge was to make the encode stage very fast, in particular there is an arithmetic divisor that works with a clock period of 9.5 ns, but since the value of divisor depends on the result of the division of the previous calculation, it is not possible to take full advantage of the divisor pipeline, whereby it needs to wait three clock cycles to complete calculation cycle.
Output bytes from Encode Symbol block are bufferized using a FIFO of 16Kbytes for each SM. When the FIFO goes full, it is emptied and all data are packetized and transmitted via channel link (see figure 8). Figure 12 illustrates the structure of each packet. The header consists of a Statistical Model Identifier Code and a Packet Data Size.
The data rate reached from the compressor system described in this paper is 20Mpix/s with a pixel depth of 16bits. The data rate is the same both in lossless and in lossy mode.

At the moment of writing the present paper, the Rate Control Unit, which allows to control the output data rate, is under finalization. The data rate of outputs compressed stream is basically variable, but making use of the Rate Control Unit it is possible archive an almost constant output data rate by a dynamic adjusting of the quantization step of Quantizer thanks to loop feedback shown in the figure 8. The target rate is configurable by user via CAN interface. Since the Rate Control Unit acts on the Quantizer, it operates only in lossy mode.

The Table 1 summarizes the FPGA's resources used by the implemented Soft IP core excluding the Rate Control Unit.

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	5,992	81,920	7%	
Number of Slice LUTs	10,254	81,920	12%	
Number of occupied Slices	4,318	20,480	21%	
Number of LUT Flip Flop pairs used	12,940			
Number of bonded IOBs	533	840	63%	
Number of BlockRAM/FIFO	73	298	24%	
Total Memory used (KB)	2,358	10,728	21%	
Number of BUFG/BUFGCTRLs	7	32	21%	
Number of DSP48Es	115	320	35%	
Number of PLL_ADVs	1	6	16%	
Average Fanout of Non-Clock Nets	3.00			

Table 1. Soft IP used resources

TEST ENVIRONMENT

Figure 13 shows the test environment (TE) used by TSD to validate the implementation of the compressor.

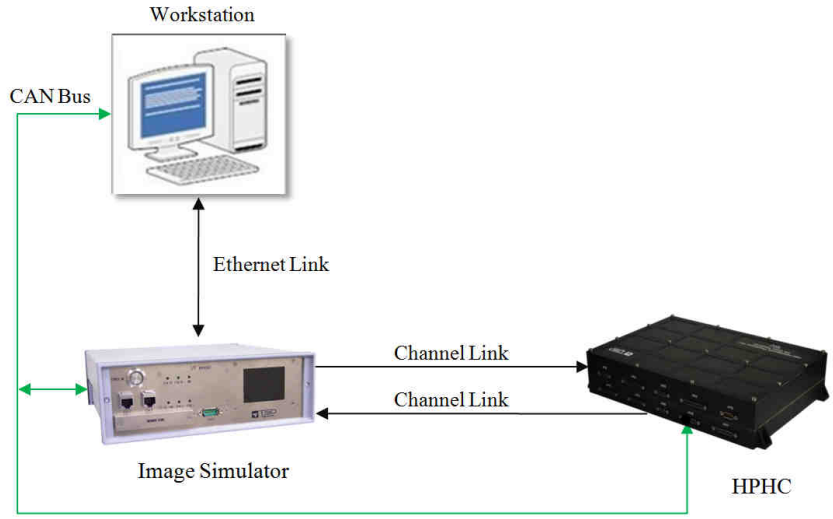

Figure 13: Test Environment

Cubes of images (COI) in BIL format are transmitted from the workstation (WS) to the Simulator via Ethernet Link. The simulator transmits sequentially the received image data to the HPHC via Channel Link. The HPHC receives the images, performs the real time compression and re-transmits the data through the Channel Link to the Simulator. Finally the Simulator transfers the data via Ethernet to the WS where it is stored and compared against the expected data.
Several parameters necessary to process the COI as the number of rows, columns and bands, as well as the parameters to setup the compressor are transmitted via CAN Bus from WS to the Simulator and HPHC.
The validation has been performed by means of a comparison between the data compressed transmitted by HPHC and the output data generated by the reference SW model written in C language. Eleven different COI have been used to test and to validate the implementation of the compression algorithm. Table 2 illustrates the image files in question and their dimensions.

COI name	rows	columns	bands
airs_gran9.raw	135	90	1501
CASI-t0477f06-nuc.raw	1225	406	72
coast.raw	1024	1024	6
m3globala.raw	512	320	86
MODIS-MOD01_500m-nuc.raw	4060	2708	5
montpellier.raw	224	2456	4
montpellier_crop_Deca-050.raw	2448	296	4
mountain.raw	1024	1024	6
SFSI_mantar_Raw.raw	140	496	240
t0477f06_rad_rss.raw	1225	406	72

Table 2. Images involved

IMAGER AND MASS MEMORY SIMULATOR

The Imager and Mass Memory Simulator is an equipment designed and manufactured by TSD to implement an Hyperspectral data generator and a real time storage of the compressed data, so to provide a real time closed loop test environment.

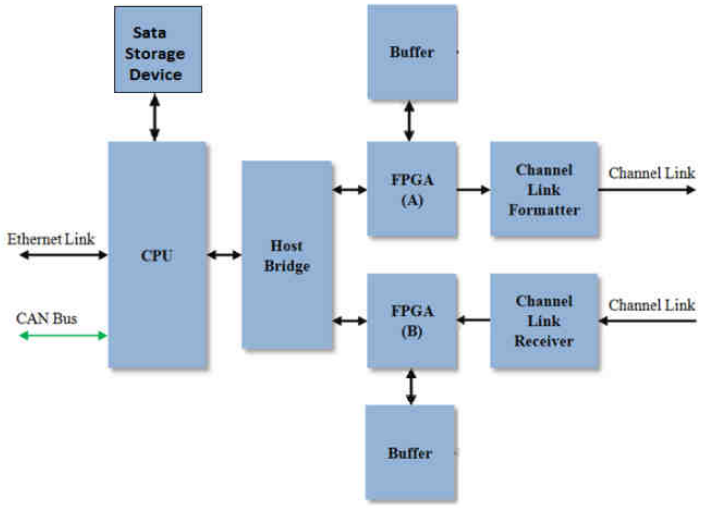

Figure 14: Basic block diagram of Simulator

The Simulator is based on a processor IBM PPC750CL PowerPC running at 600 MHz and two Virtex-4 FPGAs for image data handling at high data rate and transmission/reception over Channel Link. Two FPGAs are provided with 6 SDRAMs (64 Mbytes each) for temporary buffering of the video data streams (both uncompressed and compressed).
The sustained data transfer over Ethernet between the Simulator and the WS is 40 Mbytes/s in both directions.

CONCLUSIONS

The proposed HW implementation for onboard compression of multispectral and hyperspectral images achieves good performances in terms of maximum throughput (20Mpixels/s with a pixel depth of 16bits), good flexibility being able to process any COI with a maximum size of 4096 columns and bands and with an unlimited number of rows.
The implemented Soft IP Core requires a limited amount of HW resources of a FPGA Virtex-5 XC5VFX130T that means a possible future enhancement to integrate additional functionalities
Actually it is under finalization an enhanced version of the compressor with a Rate Control Unit which will allow the usage of the compressor also in the case in which a fixed rate is required.

REFERENCES

[1] Jyotika Doshi and Savita Gandhi, “Implementing a Novel Data Structure for Maintaining Cumulative Frequency of Symbols”, International Journal of Computer Applications (0975 – 8887) - Vol. 41 - No.15, March 2012
[2] CCSDS, "Lossless Multispectral & Hyperspectral Image Compression", CCSDS 123.0-B-1, May 2012
[3] D. Titomanlio et al., "MASER 12 Digital Video System", 20th ESA Symposium on European Rockets and Balloon Related Research. ESA SP-700 Proceedings (2011), Noordwijk, The Netherlands
[4] G. Capuano et al. "High Data Rate Image Compression HW Platforms", 64th International Astronautical Congress 2013
[5] A. Abrardo et al, ““Low-complexity approaches for lossless and near-lossless hyperspectral image compression,” Satellite Data Compression, B. Huang (Editor), Springer, Sept. 2011.
[6] I. Blanes, J. Serra-Sagristà, “Pairwise orthogonal transform for spectral image coding,” IEEE Transactions on Geoscience and Remote sensing, v. 49, n. 3, pp. 961- 972, March 2011.
[7] A. Abrardo et al., “Low-complexity predictive lossy compression of hyperspectral and ultraspectral images,” Proc. of IEEE ICASSP, 2011.
[8] CCSDS 123.0-B-1, "Lossless Multispectral & Hyperspectral Image Compression", May 2012
[9] M. De Nino et al., "Lossy Multi/Hyperspectral Compression Hw Implementation At High Data Rate", IAC 2014, Toronto