Removing Data Dependencies in the CCSDS 123.0-B-2 Predictor Weight Updating

Yubal Barrios, Joan Bartrina-Rapesta, Miguel Hernández-Cabronero, Antonio Sánchez, Ian Blanes, Joan Serra-Sagristà, *Senior Member, IEEE*, and Roberto Sarmiento

***Abstract*—The Consultative Committee for Space Data Systems (CCSDS) first standardized near-lossless coding capabilities in the CCSDS 123.0-B-2 algorithm. However, this standard does not describe strategies to produce high-throughput hardware implementations, which are not trivial to derive from its definition. At the same time, throughput optimizations without significant compression performance penalty are paramount to enable real-time compression on-board next-generation satellites. This work demonstrates that the weight update stage of the CCSDS 123.0-B-2 predictor can be selectively bypassed to enhance throughput for both lossless and near-lossless modes with minimal impact on compression performance and still produce fully compliant bitstreams. Skipping the weight update implies that those weights must be carefully chosen outside the original CCSDS 123.0-B-2 pipeline. Two strategies are proposed to select effective weight values based on whether a priori information about the current image is exploited or not. Comprehensive experimental results are presented for both proposed strategies and for lossless and near-lossless regimes, using a representative set of hyperspectral images. The coding penalty is, on average, 1% for lossless and 8% for near-lossless, depending on the strategy used to set the initial weights. The proposed method obtains a maximum throughput of 1 processed sample per clock cycle when it is evaluated using High-Level Synthesis (HLS), consuming 4.6% of the LUTs and 31.1% of the internal memory on a Xilinx Kintex UltraScale space-grade FPGA.**

***Index Terms*—Hyperspectral imaging, compression algorithms, CCSDS 123.0-B-2, on-board data processing, high throughput.**

I. Introduction

THE space industry is increasingly embracing high-resolution imaging sensors for deployment on-board satellites. Data acquired by these sensors are invaluable for multiple remote sensing applications, such as natural resource management and identification, surveillance and defense [1], [2]. Large increments in the spatial and spectral resolution of on-board sensors are expected in the next decade [3], [4], so the increasing amount of acquired data and both the computational and transmission limitations on-board satellites make necessary the use of efficient data compression techniques.

This work has been supported by ESA (contract 4000136723/22/NL/CRS), by the European Union's Horizon 2020 research and innovation programme (grant agreement No 776151), the Marie Skłodowska-Curie (grant agreement #801370), by the Spanish Ministry of Science and Innovation and by the European Regional Development Fund (PID2021-125258OB-I00), and by the Government of Catalonia (2018-BP-00008 and SGR2021-00643).

Y. Barrios, A. Sánchez and R. Sarmiento are with the Institute for Applied Microelectronics (IUMA), University of Las Palmas de Gran Canaria (ULPGC), 35017 Las Palmas de Gran Canaria, Spain (e-mail: {ybarrios, ajsanchez, roberto}@iuma.ulpgc.es).

J. Bartrina-Rapesta, M. Hernández-Cabronero, I. Blanes and J. Serra-Sagristà are with the Universitat Autònoma de Barcelona, 08193 Cerdanyola del V., Spain (e-mail: {joan.bartrina, miguel.hernandez, ian.blanes, joan.serra}@uab.cat).

Manuscript received XXX; revised XXX.

The Consultative Committee for Space Data Systems (CCSDS) has published some standards to compress different types of data acquired on space missions. The CCSDS 123.0-B-2 "Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression" standard (from now on CCSDS 123.0-B-2) [5] is the latest CCSDS definition for image data coding. CCSDS 123.0-B-2 is designed for compressing multi- and hyperspectral images on-board satellites in lossless and near-lossless regimes. Thus, it employs a low-complexity algorithm that achieves high compression ratios and controls the maximum error produced between the reconstructed and the original image. This standard yields high compression ratios by exploiting correlation between adjacent spatial and spectral samples. This introduces a considerable number of data dependencies. These dependencies constrain the throughput and the area footprint and complicate the design of hardware implementations. Efforts are being made to devise low-complexity implementations that efficiently compress the huge amount of data acquired in real-time with CCSDS 123.0-B-2 standard [6], [7] and its predecessor CCSDS 123.0-B-1, focusing on obtaining high throughputs [8], [9], [10] (one sample compressed per clock cycle, when processing in Band-Interleaved by Pixel (BIP) order, the most favourable in lossless mode).

This paper proposes a specific parameter configuration that allows bypassing the weight updating of the prediction stage of CCSDS 123.0-B-2, which entails a reduction of hardware occupancy and an increment in throughput, especially relevant when working in near-lossless mode. However, not updating the weights during the prediction stage produces larger prediction errors, which can result in a compression penalty with respect to the default prediction behaviour. This compression penalty should be assessed to guarantee mission goals.

The bypass proposal was first introduced in [11], where our initial analysis hinted at little compression penalties for data from four acquisition sensors. In the current contribution, we fully analyse the bypass approach for the complete set of sensors included in the data set defined by the CCSDS [12]. Moreover, we modify an existing CCSDS 123.0-B-2 implementation to include the proposed technique and present a hardware comparison of its resource usage and performance. This letter employs the same notation as the CCSDS 123.0-B-2 standard. Readers not familiar with this standard may obtain additional information from [5], [6], [13].

The rest of the paper is organized as follows. Section II details the proposed method for selective weight update bypass and describes the two strategies proposed to select the initial weights. Section III provides a comprehensive performance

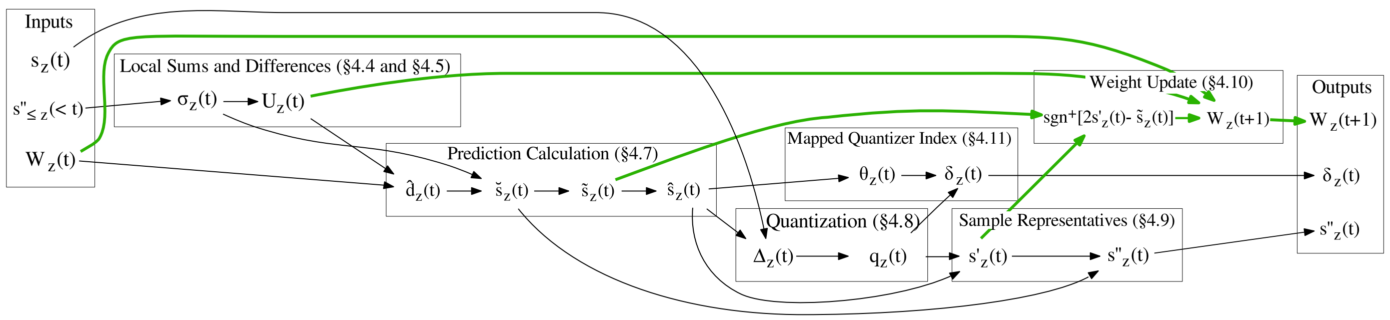

Fig. 1: Data dependency diagram for the CCSDS 123.0-B-2 predictor. The critical path bypassed by the proposed technique is highlighted in green.

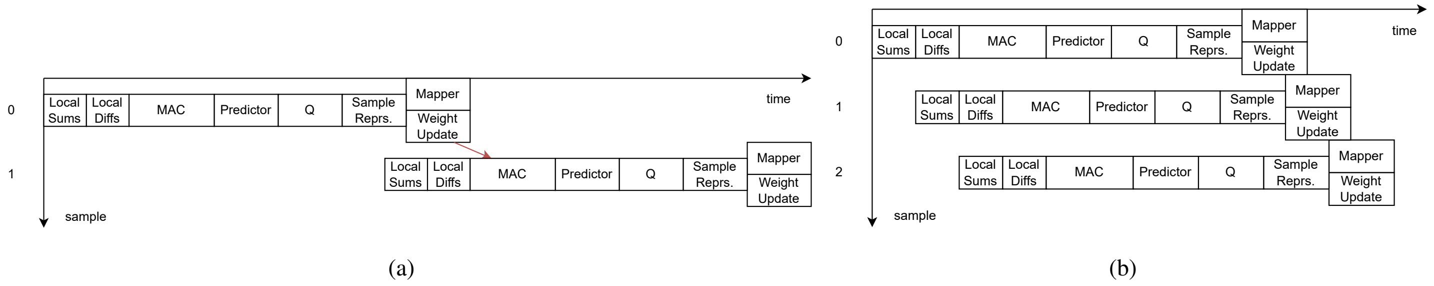

Fig. 2: Pipeline disposition (a) when processing image samples using narrow local sums and reduced prediction, and (b) when the weight updating is bypassed.

analysis, including throughput and resource utilization results for a representative space-grade Field Programmable Gate Array (FPGA). Finally, Section IV draws the main conclusions of this work.

II. CCSDS 123.0-B-2 Weight Update Bypass

As thoroughly explained in [6], sample representative computation in the CCSDS 123.0-B-2 predictor introduces data dependencies when processing consecutive samples in near-lossless mode, incurring in a penalty in terms of hardware performance. Data dependencies vary depending on the input image order, resulting in different spectral and spatial relationships among samples. Although these data dependencies can be partially alleviated by selecting appropriate options (e.g., narrow local sums together with reduced prediction mode), there is still margin to further suppress these data dependencies, increasing the throughput and reducing hardware occupancy. Fig. 1 depicts data dependencies of the CCSDS 123.0-B-2 predictor stage.

Within the CCSDS 123.0-B-2 predictor, one of the most computationally expensive operations is the weight update stage, which requires a dot product to get the weight vector $\omega_z^i(t+1)$ for the sample $t+1$ from $\omega_z^i(t)$, where i denotes N, W, and NW neighbours of the current pixel, z the band under encoding, and t the spatial index of the current pixel. This step, commonly implemented on hardware using a multiply & accumulate (MAC) unit, entails strong data dependencies in the processing of two consecutive input samples. The path bypassed with the proposed parameter set configuration is highlighted in green in Fig. 1. However, properly selecting a reduced set of parameters bypasses the weight update stage, enabling faster and smaller hardware implementations, and even completely removes data dependencies between consecutive samples in Band-Interleaved by Line (BIL) order. This order is the one employed by pushbroom sensors, which are the most common for spectroscopic imagery.

A. Constant Weight Condition

When some parameters of the CCSDS 123.0-B-2 predictor ($\zeta_z^{(i)}$, ν_{min}, ν_{max}. and Ω) are properly set, weight update stage (defined in Section 4.10 of [5]) is skipped. It is proven in [11] that when $\zeta_z^{(i)} = \zeta_z^* = 5$, $\nu_{\text{min}} = \nu_{\text{max}} = 9$, and $\Omega \le 11$ the CCSDS 123.0-B-2 predictor operates under constant weights. Thus, $\omega_z^i(t) = \omega_z^i(t+1)$ and weight updating is bypassed. This is hereafter referred to as *Fast Mode*. Therefore, for this mode, weights are not learned, but must be properly defined previously. To do this, two different strategies are described later.

Other values of $\zeta_z^{(i)}$, ζ_z^*, ν_{min}, and ν_{max} may still achieve the same effect, since they do not have any other effect on compression performance and the current selection allows larger values of Ω. Weights have a resolution of $\Omega+3$ bits. Hence, weights with 14-bit resolution can be employed even if the weight update is bypassed and still produce compliant bitstreams. It is worth noting that higher weight resolution produces better estimations and thus better coding performance. Using the mentioned parameter configuration might produce implementations where critical data dependencies within a single image line, i.e., under BIL order, are completely removed, moving from a predictor implementation with strong data dependencies to a fully pipelined approach. We outline that, when the weight update bypass is combined with narrow local sums and reduced prediction mode, the maximum throughput achieved is of 1 sample processed per clock cycle. Fig. 2 shows the pipelined approach with strong dependencies and the proposed one.

B. Weight Generation Strategies

It is remarkable that this predictor configuration necessarily relies on a well-crafted set of initial weights. Otherwise, the compression ratio is significantly penalized. Two strategies are presented here, while the encoding penalty derived from using constant weights is analyzed in Section III.

The CCSDS 120.2-G-2 Informational Report [14] recommends a default set of parameters for compressing scenes depending of the acquisition sensor. When default configuration is employed, the weight vector $\omega_z^i(t)$ is updated after coding a new sample. From now on we name this strategy as *Default Mode*. On the other hand, and for the *Fast Mode*, two different strategies to set $\omega_z^i(t)$ are described next:

- ***Partial Updating*** codes the first N'_Y lines with the *Default Mode*, and the remaining $N_Y - N'_Y$ with the *Fast Mode*, where $N'_Y > 0$ is a user-defined parameter.
- ***Exogenous*** uses the *Fast Mode* for all lines of the image using previously computed weights. These weights are computed applying the *Default Mode* to one or more scenes of the same sensor.

Note that both strategies can be used with or without knowledge of the sensor properties and produce a compliant bitstream, because the CCSDS 123.0-B-2 standard allows the manual updating of an implementation employed weights. Thus, for *Partial Updating* after coding N'_Y lines, the headers are configured to *Fast Mode*, while for *Exogenous* the weights are computed asynchronously and the whole image is compressed in *Fast Mode*.

III. Experimental Results

This section analyzes the impact on compression performance of the *Fast Mode* when employing *Partial Updating* and *Exogenous* strategies for computing the initial and fixed weights, compared with using the *Default Mode* for the whole image. Besides, a comparison in terms of throughput and area utilization is provided for different FPGA implementations of the CCSDS 123.0-B-2 standard, including the one proposed in this work.

A. Compression Performance Analysis

Compression performance is measured, in terms of bit per sample (bps), by employing a software implementation of the CCSDS 123.0-B-2 standard [15] for the *Default* and *Fast Mode* configurations. Standard hardware implementations produce the same coding performance.

With the aim of exhaustively evaluating the coding performance, results are obtained for all scenes of the 12 different sensors of the test corpora gathered by the CCSDS Data Compression working group [14], [12]. This is a well-known dataset and deeply used for algorithmic evaluation on hyperspectral imaging processing. Image characteristics such as the number of scenes per sensor, the number of bands (N_z), rows (N_y) and columns (N_x) per image, binary entropy in bps, and the amount of data collected per sensor in Megabytes (MB) are summarized in Table I.

TABLE I: Image Corpus.

Sensor	Scenes	Dimensions ($N_z \times N_y \times N_x$)	Entropy (bps)	Size (MB)
Airs	9	(1501×135×90)	11.32	313
AVIRIS	1	(224×512×614)	8.55	134
	3	(224×512×680)	11.34	447
CASI	1	(72×2852×405)	10.39	159
	1	(72×1225×406)	10.66	68
CRISM	7	(74×2700×64)	9.63	168
HICO	3	(87×2000×500)	11.10	498
	2	(128×2000×512)	10.36	500
Hyperion	1	(242×3187×256)	9.46	377
	1	(242×3176×256)	9.35	375
	1	(242×3242×256)	9.91	383
Landsat	3	(6×1024×1024)	6.64	36
M3	2	(86×512×320)	9.36	54
	3	(86×512×640)	9.07	489
MODIS	5	(14×1354×2030)	7.23	365
	3	(17×1354×2030)	10.75	267
Pleiades	1	(4×2456×224)	10.84	4
	4	(4×2448×296)	10.18	22
	1	(4×3928×224)	10.57	7
Spot5	3	(3×1024×1024)	6.83	18
Vegetation	2	(4×10080×1728)	9.41	266
	2	(17×10193×1728)	9.41	268
Total	**58**	–	**9.62**	**5118**

The compressor configuration parameters for the *Default Mode* are those defined in [14] to maximize the compression ratio. For the *Fast Mode*, the parameters are also the ones defined in [14], except for those related to the weight updating, which are the ones defined in Section II-A. For near-lossless compression results, the band-independent absolute error limit assignment method is employed.

The first experiment evaluates the compression performance impact of *Fast Mode* using the *Partial Updating* strategy. Average results for all scenes of each sensor under study are shown in Fig. 3. In these plots, the vertical axis provides the overhead in % (smaller is better) of employing the *Partial Updating* strategy as compared to the use of *Default Mode* for the whole scene. From this experiment, it can be observed that after coding around 100 lines in *Default Mode*, the compression overhead is quite stable for all sensors, and for all tested Peak Absolute Errors (PAEs). The time needed to reach the coding performance stabilization point depends on the image size and the N'_Y value. It is interesting to note that, for some sensors, the impact in the coding performance at different PAEs is nearly insignificant, e.g., for AVIRIS, CASI, MODIS, Pleaiades and Spot5. However, for Airs, CRISM, HICO, Hyperion, Landsat, M3 and Vegetation, the coding penalty is larger for higher values of PAE. A particular case is observed for Landsat, which obtains a coding performance up to 10% better than coding the whole image in *Default Mode*. This effect would suggest that the predictor does not estimate the scanned data because of the poor spatial resolution of the sensor.

In addition, some peaks are observed in the coding overhead, which occur due to weights set during the training process. Since the training process when selecting the Partial Updating strategy uses the first N'_Y lines, if the last $N_Y - N'_Y$ lines have very different statistics from the rest of the image, the weights are updated accordingly and therefore the prediction will not be accurate, increasing the penalty in terms of coding performance.

The second experiment evaluates the *Fast Mode* using *Exogenous* strategy. This strategy requires a training image with the same number of columns than the scenes where the weights will be finally used. This limits the amount of sensors to be analyzed for this strategy. The scenes

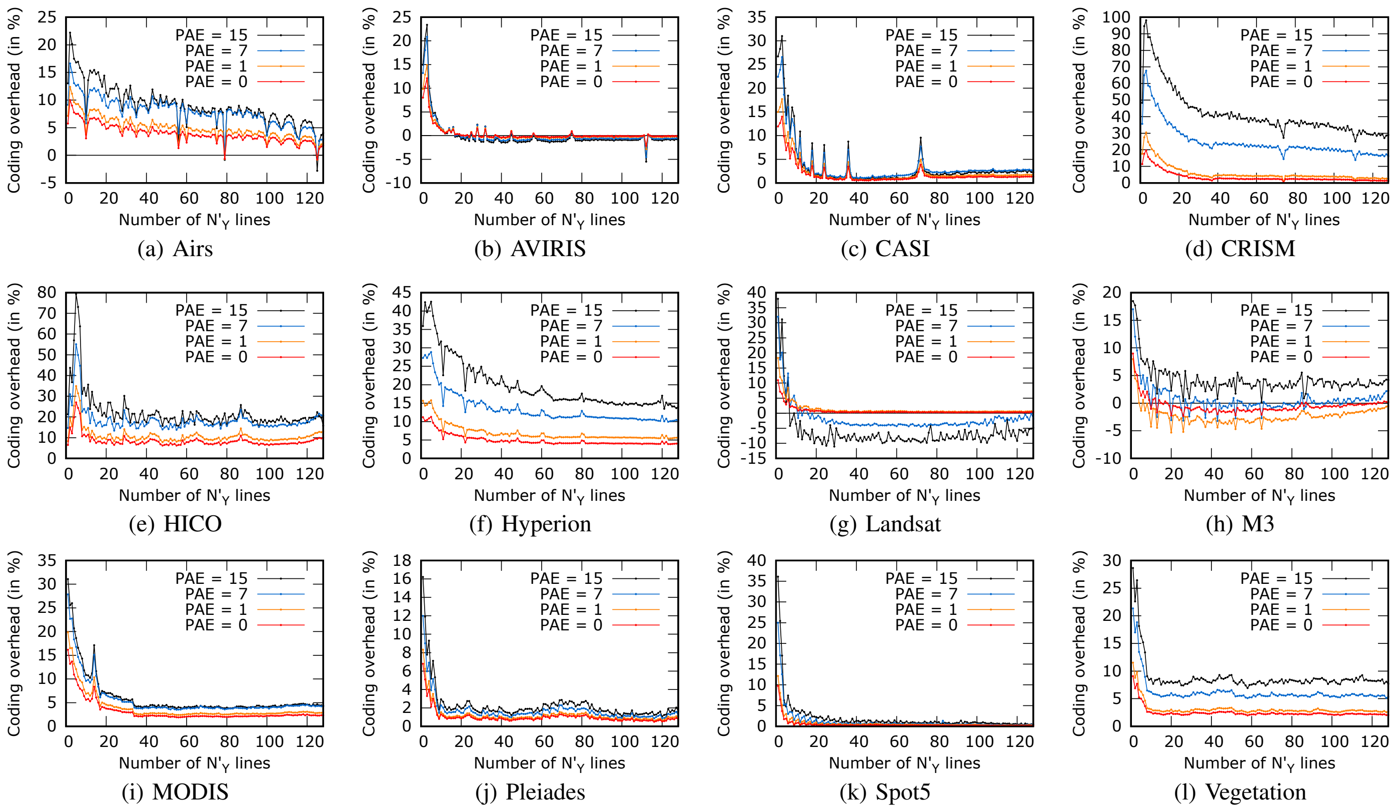

Fig. 3: Partial Updating coding performance evaluation.

TABLE II: Coding performance evaluation of the Exogenous strategy, measured in terms of bits per sample (bps).

PAE	Airs	AVIRIS	CRISM	HICO	Hyperion	Landsat	M3	MODIS	Pleiades	Spot5	Vegetation
0	7.60 (2.15%)	9.71 (-1.22%)	4.05 (-0.49%)	7.43 (5.09%)	5.59 (5.47%)	4.09 (0.00%)	4.56 (-6.94%)	8.72 (1.40%)	7.83 (1.03%)	4.88 (0.21%)	6.45 (4.2%)
1	6.03 (2.55%)	8.14 (-1.33%)	2.58 (-1.53%)	5.86 (6.55%)	4.00 (6.95%)	2.50 (0.40%)	3.21 (-13.48)	7.16 (1.70%)	6.26 (1.29%)	3.48 (0.29%)	4.90 (5.15%)
3	4.89 (1.87%)	6.90 (-1.99%)	1.61 (-1.23%)	4.72 (7.76%)	2.86 (9.58%)	1.48 (2.07%)	2.22 (-12.94%)	6.00 (2.21%)	5.06 (1.20%)	2.23 (0.45%)	3.80 (6.44%)
7	3.89 (3.18%)	5.80 (-2.68%)	0.95 (0.00%)	3.74 (11.31%)	1.95 (14.71%)	0.76 (-8.43%)	1.52 (-7.32%)	4.97 (2.05%)	4.04 (1.51%)	1.33 (0.76%)	2.80 (10.67%)
15	2.89 (3.21%)	4.82 (-2.63%)	0.53 (1.92%)	2.71 (15.81%)	1.18 (15.69%)	0.34 (-17.07%)	0.95 (-7.77%)	4.05 (1.76%)	3.02 (2.37%)	0.67 (0.00%)	1.89 (16.67%)
31	1.95 (3.17%)	3.95 (-1.74%)	0.29 (7.41%)	1.69 (11.18%)	0.71 (24.56%)	0.10 (-44.44%)	0.55 (-9.84%)	3.20 (2.56%)	2.08 (3.48%)	0.25 (-13.79%)	1.21 (27.37%)

employed for generating the weights with the *Default Mode* are ErtaAleEO1H1680502010057110KF, gran120, maine_f030828t01p00r05_sc10_uncal, A2001123_0000day, msp00003e34_01_sc214, AquaAlta, agriculture, targetA, montpellier, toulouse, and 1_1b for the sensors Hyperion, Airs, AVIRIS, MODIS, CRISM, HICO, Landsat, M3, Pleiades, Spot5, Vegetation, respectively. Table II shows the compression performance in bps and the overhead (in %) with respect to compressing the whole scenes in *Default Mode*. Results show overheads that are, on average, of 0.99%, 0.78%, 1.40%, 2.34%, 2.72% and 0.90% when the images are compressed at PAEs of 0 (i.e., lossless regime), 1, 4, 7, 15, and 31, respectively. Results also show that for sensors CRISM, HICO, Hyperion and Vegetation, the encoding penalty is significant, reaching approximately 25% with a PAE of 31. On the other hand, for the other evaluated sensors, the coding penalty is lower than 5%, and in some cases even produces better coding performance.

B. Hardware Evaluation

The implications of bypassing the weight update stage are also analyzed from the point of view of a hardware implementation. This analysis considers both throughput and hardware occupancy as relevant metrics. FPGA occupancy is measured in terms of Look-Up Tables (LUTs), Flip-Flops (FFs), and memory resources (BRAMs), as well as the usage of dedicated modules for complex arithmetic operations (DSPs). The target sensor for the reported results is AVIRIS with images segmented every 512 lines (224 bands × 677 samples × 512 lines, 16 bits per pixel). To do this analysis, the High-Level Synthesis (HLS) implementation of the CCSDS 123.0-B-2 standard presented by the authors in [16] is taken as starting point. This implementation has been modified not only to evaluate the proposed *Fast Mode*, but also to obtain HLS results of the optimization strategies presented in [6] (from here on *High-Performance* approach). Results for both approaches are obtained considering only the prediction stage of CCSDS 123.0-B-2, since it is the main bottleneck in a hardware implementation of the full compressor. These results are shown in Table III, in which they are compared to other available hardware implementations of the CCSDS 123.0-B-2 standard. In the case of [16], results are shown for the whole compressor and also considering just the prediction stage. Please note that the implementation proposed in [7] is the only one directly implemented at Register-Transfer Level

TABLE III: Area and throughput comparison for different CCSDS 123.0-B-2 FPGA implementations

Implementation	Methodology	Order	Encoder	Device	LUTs	FFs	DSPs	BRAMs	Freq. (MHz)	Throughput (MSamples/s)
Barrios et al. [16]	HLS	BIL	Hybrid	Kintex XCKU040	17185	11915	63	85	125	17.86
Barrios et al. [16]	HLS	BIL	N/A	Kintex XCKU040	10087	6054	54	76	125	17.86
Bascones et al. [7]	RTL	FID	Hybrid	Kintex XQRKU060	16458	15707	30	187.5	250	249.6
High-Performance [6]	HLS	BIL	N/A	Kintex XCKU040	19840	12970	33	213	230.5	230.5
Fast Mode (proposed)	HLS	BIL	N/A	Kintex XCKU040	11095	11729	20	186.5	235.5	235.5

(RTL), hence a more optimized implementation in terms of timing and hardware occupancy is expected.

As it can be observed, using either the *High-Performance* or the *Fast Mode* approach clearly speeds-up the hardware implementation when compared to [16], obtaining in both cases a maximum throughput of 1 sample processed per clock cycle. This is directly achieved in BIL order, without applying any prior reordering stage. This means that it is required to store about one spectral line ($N_xN_z + 1$) of sample representatives, N_x local difference vectors and N_z weight vectors. Reordering is required in [7], transforming from BIP order to a new proposed one named Frame-Interleaved by Diagonal (FID). FID order is based on unrolling a hyperspectral image into a 2D plane constructed by appending consecutive image bands and then sweeping that plane diagonally. This allows avoiding data dependencies at the expense of introducing a high latency during the reordering from BIP to FID, which is even higher when the input image arrangement is BIL or BSQ. Even implementing the proposed strategies using HLS, the penalty in terms of throughput is reduced when compared to [7] (a maximum reduction of about 7.7% and 5.6% regarding the *High-Performance* and the *Fast Mode* approaches, respectively).

In contrast, both the *High-Performance* and *Fast Mode* approaches have a higher logic resources utilization than the predictor presented in [16] even when the weight update stage is omitted. This is because of the parallel processing between consecutive samples in absence of data dependencies. This penalty is higher in the case of the *High-Performance* approach, with an increment of about 96.7% of LUTs and 114.2% of FFs compared to [16]. Regarding [7], similar memory consumption (in terms of BRAMs usage) is appreciated when implementing either the *High-Performance* or the *Fast Mode* approach. On the other hand, a clear reduction in terms of DSPs and LUTs (30% and 32.6%, respectively) is observed when comparing the *Fast Mode* approach to [7], mainly related to the absence of the weight updating stage, which performs some complex arithmetic operations, such as multiplications.

IV. Conclusion

In this work, a *Fast Mode* for the CCSDS 123.0-B-2 predictor is presented. It is based on selectively bypassing the weight update stage by properly tuning some user-defined parameters. In this way, the compression is accelerated at the expense of some compression performance penalty. Two strategies to apply the *Fast Mode* have been presented. In both cases, the coding overhead is between 1% and 5% for the lossless regime. In the near-lossless regime, this penalty increases proportionally to the quantization step, with overheads in the range from 3% to 25%. It is worth noting that, in some cases, some coding savings are even achieved. Furthermore, the *Fast Mode* is able to perform on-board hyperspectral image compression in real-time with a restrained area footprint, allowing performance figures comparable to other hardware throughput-optimized CCSDS 123.0-B-2 based implementations.

References

[1] S.-E. Qian, "Introduction to hyperspectral satellites," in *Hyperspectral Satellites and System Design*. CRC Press, Taylor & Francis Group, 2020, ch. 1, pp. 1–52.

[2] S. Peyghambari and Y. Zhang, "Hyperspectral Remote Sensing in Lithological Mapping, Mineral Exploration, and Environmental Geology: an Updated Review," *Journal of Applied Remote Sensing*, vol. 15, no. 3, p. 031501, 2021. [Online]. Available: https://doi.org/10.1117/1.JRS.15.031501

[3] G. Denis, A. Claverie, X. Pasco, J.-P. Darnis, B. de Maupeou, M. Lafaye, and E. Morel, "Towards Disruptions in Earth observation? New Earth Observation systems and markets evolution: Possible scenarios and impacts," *Acta Astronautica*, vol. 137, pp. 415–433, 2017.

[4] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern Trends in Hyperspectral Image Analysis: A Review," *IEEE Access*, vol. 6, pp. 14 118–14 129, 2018.

[5] Consultative Committee for Space Data Systems, *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression, CCSDS 123.0-B-2*. CCSDS, February 2019, vol. Blue Book.

[6] A. Sánchez, I. Blanes, Y. Barrios, M. Hernández-Cabronero, J. Bartrina-Rapesta, J. Serra-Sagristà, and R. Sarmiento, "Reducing Data Dependencies in the Feedback Loop of the CCSDS 123.0-B-2 Predictor," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[7] D. Báscones, C. Gonzalez, and D. Mozos, "A Real-Time FPGA Implementation of the CCSDS 123.0-B-2 Standard," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[8] Y. Barrios, A. Sánchez, L. Santos, and R. Sarmiento, "SHyLoC 2.0: a versatile hardware solution for on-board data and hyperspectral image compression on future space missions," *IEEE Access*, vol. 8, pp. 54 269–54 287, 2020.

[9] J. Fjeldtvedt, M. Orlandić, and T. A. Johansen, "An Efficient Real-Time FPGA Implementation of the CCSDS-123 Compression Standard for Hyperspectral Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 10, pp. 3841–3852, 2018.

[10] A. Tsigkanos, N. Kranitis, G. Theodorou, and A. Paschalis, "A 3.3 Gbps CCSDS 123.0-B-1 Multispectral & Hyperspectral Image Compression Hardware Accelerator on a Space-Grade SRAM FPGA," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 90–103, 2021.

[11] J. Bartrina-Rapesta, M. Hernández-Cabronero, A. Sánchez, Y. Barrios, R. Sarmiento, J. Serra-Sagristà, and I. Blanes, "Coding Performance Impact on a Constant-Weight Predictor for CCSDS 123.0-B-2," in *On-Board Payload Data Compression (OBPDC), Athens, Greece*. Zenodo, Sep. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7110999

[12] Test Corpora Defined by the CCSDS Data Compression Working Group. [Online]. Available: https://cwe.ccsds.org/sls/docs/SLS-DC/123.0-B-Info/TestData/

[13] M. Hernández-Cabronero, A. B. Kiely, M. Klimesh, I. Blanes, J. Ligo, E. Magli, and J. Serra-Sagristà, "The CCSDS 123.0-B-2 "Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression" Standard: A comprehensive review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 102–119, 2021.

[14] Consultative Committee for Space Data Systems, *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression, CCSDS 120.2-G-2, Informational Report*. CCSDS, December 2022, vol. Green Book.

[15] (2022, May) CCSDS 123.0-b-2 & CCSDS 121.0-b-3 Hyperspectral image compression algorithms. [Online]. Available: https://www.connectbycnes.fr/en/ccsds-1230-b-2-ccsds-1210-b-3

[16] Y. Barrios, A. Sánchez, R. Guerra, and R. Sarmiento, "Hardware Implementation of the CCSDS 123.0-B-2 Near-Lossless Compression Standard Following an HLS Design Methodology," *Remote Sensing*, vol. 13, no. 21, p. 4388, 2021.