The CCSDS 123.0-B-2 Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression Standard: A comprehensive review

The CCSDS 123.0-B-2 "Low-complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression" standard, explained

Miguel Hernández-Cabronero*, Aaron B. Kiely, *Senior Member, IEEE*, Matthew Klimesh, *Senior Member, IEEE*, Ian Blanes, *Senior Member, IEEE*, Jonathan Ligo, *Member, IEEE*, Enrico Magli, *Fellow, IEEE*, Joan Serra-Sagristà, *Senior Member, IEEE*

***Abstract*—The Consultative Committee for Space Data Systems (CCSDS) published the CCSDS 123.0-B-2 compression standard for multispectral and hyperspectral images. This standard extends the previous Issue, CCSDS 123.0-B-1, which only supported lossless compression, while maintaining backward compatibility. The main novelty of the new Issue is support for near-lossless compression, i.e., lossy compression with user-defined absolute and/or relative error limits in the reconstructed images. This new feature is achieved via closed-loop quantization of prediction errors. Two further additions arise from the new near-lossless support: first, the calculation of predicted sample values using sample representatives that may not be equal to the reconstructed sample values; second, a new hybrid entropy coder designed to provide enhanced compression performance for low-entropy data, prevalent when non-lossless compression is used. These new features enable significantly smaller compressed data volumes than those achievable with CCSDS 123.0-B-1, while controlling the quality of the decompressed images. As a result, larger amounts of valuable information can be retrieved given a set of bandwidth and energy consumption constraints.**

***Index Terms*—image compression, near lossless, hyperspectral imaging, remote sensing, compression standards.**

I. Introduction

During the past 30 years, multispectral and hyperspectral imaging (HSI) has become a staple tool for geoscience remote sensing and Earth observation [1], [2]. This type of imagery enables simultaneous registration of multiple parts of the electromagnetic spectrum, which provide invaluable information for many detection, classification and unmixing problems [3]. As a result, remote sensing HSI is used nowadays in many commercial, scientific, and defense areas including precision agriculture, mining, forestry, coastal and oceanic observation, intelligence, and disaster monitoring [3]–[6]. Due to the growing amount of deployed sensors [7], the number of public and private remote sensing stakeholders [4], and the ongoing effort to improve the analysis of the retrieved images [8]–[19], the importance of HSI is likely to increase in the future.

Images produced by multispectral and hyperspectral sensors consist of multiple spectral bands, instead of the three —red, green and blue— present in traditional color images. Depending on the application and the available hardware, the number of registered bands can be in the order of tens, hundreds and even thousands [20]. Thus, HSI generates significantly larger volumes of data compared to traditional imagers. Moreover, the spatial resolution of the deployed sensors also follows a rising trend, further increasing the amount of data produced. For instance, the HyspIRI sensor developed by NASA can produce up to 5 Terabytes of data per day [21]. However, the downlink channel capacity between the remote sensing devices and the ground stations is constrained, which limits the amount and quality of the retrieved data [22].

Data compression is typically applied to reduce the amount of data to be downloaded, hence improving effective transmission capacity [23]–[27]. Due to hardware and energy constraints, employed algorithms must be tailored to attain a beneficial trade-off between complexity and efficiency [22], [28]. When *lossless* compression is applied to the images, the resulting compressed data suffices to reconstruct identical copies of the originals. On the other hand, *lossy* compression enables transmitting even smaller data volumes, at the cost of reconstructed images not being identical to the originals. Among lossy compression algorithms, those that provide user-controlled bounds on the maximum error introduced in any sample are referred to as *near-lossless*.

In spite of the distortion introduced by lossy and near-lossless methods, several studies conclude that reconstructed images can be successfully used for the intended analysis tasks [29]. This is sometimes observed for compressed images up to 25 times smaller than the original ones [30]. Notwithstanding, successful analysis can only be performed when the amount of loss is adequate for the type of images and the task at hand [29], [31]. One of the main advantages of near-lossless compressors is that they offer a guarantee on

M. H.-C., I. B. and J. S.-S. acknowledge partial funding from the postdoctoral fellowship programme Beatriu de Pinós, reference 2018-BP-00008, funded by the Secretary of Universities and Research (Government of Catalonia) and by the Horizon 2020 programme of research and innovation of the European Union (EU) under the Marie Skłodowska-Curie grant agreement #801370, by EU's Horizon 2020 Program under grant agreement #776151, by the Spanish Government under grant RTI2018-095287-B-I00, and by the Catalan Government under grant 2017SGR-463. The research conducted at the Jet Propulsion Laboratory, California Institute of Technology, was performed under a contract with the National Aeronautics and Space Administration (NASA).

*M. H.-C., I. B. and J. S.-S. are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain (e-mail: miguel.hernandez@uab.cat).

A. K., and M. K. are with the Jet Propulsion Laboratory, California Institute of Technology, USA. J. L. was with the Jet Propulsion Laboratory, California Institute of Technology, USA, now with the Applied Physics Laboratory, Johns Hopkins University, USA.

E. M. is with Dept. of Electronics and Telecommunications, Politecnico di Torino, Italy.

the accuracy of all reconstructed samples in an image. This is in contrast to regular lossy compression approaches, which typically provide competitive *average* distortion results, but no assurance about the fidelity of any given set of samples. Regardless of the employed compression regime, compression algorithms must meet very stringent limitations in terms of complexity and required computational resources [32]. This constraint is particularly relevant for small satellites and CubeSats, which are attracting much scientific and industrial interest recently [4], [33].

The Consultative Committee for Space Data Systems (CCSDS), founded in 1982, publishes standards for spaceflight communication used in over 900 space missions to date.[1] CCSDS standards enable cooperation between space agencies and with industrial associates, seeking enhanced interoperability, reliability, and cost-effectiveness. The latest CCSDS compression standard is CCSDS 123.0-B-2 [34] —titled *Low-complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*, and the central topic of this paper—, which supersedes CCSDS 123.0-B-1 [35] while maintaining backward compatibility. In the CCSDS naming convention, suffixes "-1" and "-2" denote respectively the first and second issues of a standard. Hereinafter, CCSDS 123.0-B-1 and CCSDS 123.0-B-2 are also denoted as *Issue 1* and *Issue 2*, respectively.

Perhaps the most relevant novel feature of Issue 2 is a new near-lossless compression regime, enabled by a closed-loop scalar quantizer in the prediction stage [36]. Note that this in-loop quantization approach enables higher compression performance than quantization of input samples before prediction [26]. With this new feature, users can specify the maximum error limits —absolute and/or relative— introduced in the decompressed images. Fidelity settings can vary from band to band, and can be periodically updated within an image. Another new feature of Issue 2 is a *hybrid* entropy coder option. It is specifically designed to provide improved performance on low-entropy data, i.e., for the case when prediction errors tend to be small compared to the quantizer stepsize. The hybrid encoder extends the sample-adaptive codes of CCSDS 123.0-B-1 with 16 additional variable-to-variable length codes, which can represent multiple input symbols using a single codeword. To guarantee backward compatibility, both lossless and near-lossless compression can be performed with either of CCSDS 123.0-B-1's original entropy coders, or with the new hybrid option. A third novelty in the new standard is a new mode within the predictor stage called narrow local sums, designed to facilitate the design of efficient hardware implementations. Yet another change introduced in the new standard is added support for optional *supplementary information tables*, which can provide ancillary image or instrument information, e.g., to identify wavelengths associated with each spectral band.

This paper provides a comprehensive overview of Issue 2, paying special attention to new concepts and capabilities not present in Issue 1. Contents hereafter presented extend those presented in a previous conference work [36]. The following overview is more in-depth, it assumes no previous knowledge of Issue 1, and a performance evaluation is included. Furthermore, experimental results discussed here complement those in [37] by providing both a quantitative and qualitative comparison to other relevant compression methods. The rest of this paper is structured as follows: Section II describes the general structure of the compression standard, as well as its main functional parts; Section III characterizes the new standard's compression performance, based on images acquired in real space missions; finally, concluding remarks are provided in Section IV.

[1]An updated list of space missions using CCSDS standards can be found at https://public.ccsds.org/implementations/missions.aspx.

II. The new CCSDS 123.0-B-2 standard

A. Previous work

The CCSDS Data Compression (DC) working group (1995-2007, 2020-present) and the Multispectral and Hyperspectral Data Compression (MHDC) working group (2007-2020) have developed and maintained several compression standards applicable to remote sensing HSI, listed chronologically in Table I. The CCSDS 121.0-B-1 standard describes a general-purpose adaptive entropy coder. In CCSDS 121.0-B-2, the efficiency and flexibility of this entropy coder was enhanced, by allowing larger block sizes, and the possibility of using a restricted set of codewords. Since this entropy coder is available in the new CCSDS 123.0-B-2 standard, an overview is provided later in Section II-D1. The CCSDS 122.0-B-1 standard was designed specifically for image data, and supports both lossless and lossy regimes. It consists of a spatial discrete wavelet transform, which is then followed by a bit-plane coder. The CCSDS 122.1-B-0 standard extends CCSDS 122.0-B-1 by allowing the application of spectral decorrelation transforms. To provide compatibility between the 122.0 and 122.1 standards, a second issue of 122.0 (CCSDS 122.0-B-2) was also published. Finally, the CCSDS 123.0-B-1 standard formalizes a predictive coding scheme for multispectral and hyperspectral data. This standard is the immediate predecessor of the one being addressed in this tutorial, and their functional blocks are described in subsequent subsections.

Several hardware implementations can be found in the literature of the CCSDS 123.0-B-1 standard. In [44], a par-

TABLE I: Chronology of CCSDS data compression standards. Active recommendations (Blue books) are shown in blue, while retired (superseded) standards (Silver books) are shown in gray. Lossless, lossy and near-lossless compression regimes are denoted as LL, LS, and NL, respectively. The Multispectral column indicates whether several bands can be compressed simultaneously.

Name	Release	Status	Regime	Multispectral
121.0-B-1 [38]	May 1997	Retired	LL	No
122.0-B-1 [39]	May 2005	Retired	LL, LS	No
121.0-B-2 [40]	Apr 2012	Retired	LL	No
123.0-B-1 [35]	May 2012	Retired	LL	Yes
122.0-B-2 [41]	Sep 2017	Active	LL, LS	No
122.1-B-1 [42]	Sep 2017	Active	LL, LS	Yes
123.0-B-2 [34]	Feb 2019	Active	LL, NL	Yes
121.0-B-3 [43]	Aug 2020	Active	LL	No

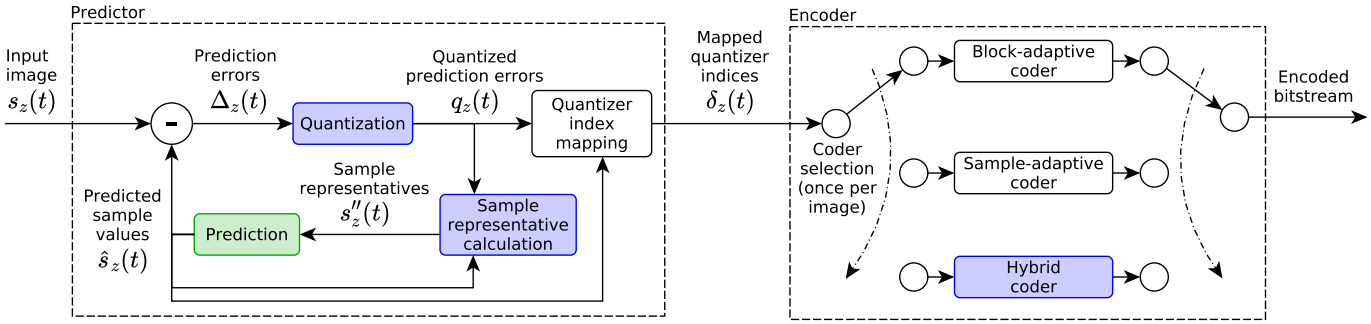


Fig. 1: Structure overview of the CCSDS 123.0-B-2 compressor. New functional blocks respect to CCSDS 123.0-B-1 are highlighted in blue, while modified blocks are shown in green.

allelization technique is described that achieves from 31 to 123 Megasamples per second (Ms/s) on the Xilinx V-7 XC7VX690T and V-5QV FX130T FPGAs, respectively. In [45], parallelization using C-slow retiming is proposed, which achieves a throughput of up to 213 Ms/s on a space-grade Virtex-5QV FPGA. In [46], another implementation is described, with a throughput of 147 Ms/s on a Xilinx Zynq-7020 FPGA. The FPGA design discussed in [47] allows parallel processing of any number of samples, provided that resource constraints are met. This enables configurable trade-offs between throughput and power consumption. In [48], a low-cost FPGA design is described for the prediction block of CCSDS 123.0-B-1, with throughput as high as 20 Ms/s on a Xilinx Zynq-7000 FPGA. In [49]–[51], low-complexity and low-occupancy FPGA designs are proposed. These implementations are designed to be independent, and combinable in a plug-and-play fashion. The latest version of this system, referred to as SHyLoC 2.0, yields a throughput of 150 Ms/s on a Xilinx Virtex XQR5VFX130 FPGA. Hardware designs for CCSDS 123.0-B-2 are currently ongoing, with the European Commission funding two research projects within the framework of the H2020 programme [52], [53], with NASA and ESA funding other research projects [54], [55]. To the best of the authors' knowledge, there are no public implementations of Issue 2 available.

Extensions to CCSDS compression algorithms have been published as well. In [56], a method to extend lossless predictive coding schemes —in particular CCSDS 123.0-B-1— was proposed. This method enables compression in lossy regime, producing constant signal-to-noise ratio and accurate rate control. In [57], a lightweight arithmetic coder was proposed as a possible replacement of the entropy coder of CCSDS 123.0-B-1. Some algorithms have been proposed related to the prediction stage of Issue 2, based on recursive least-squares theory. These algorithms describe more adaptive prediction methods, at the cost of increased computational complexity. In [58], the inverse correlation matrix of the local differences is used to update the prediction weights. In [59], this predictor is enhanced by adaptively selecting the number of local differences to be used. In [60], two prediction modes are described: the first one uses only spectral neighbors in the weight update process, the second one also employs spatial neighbors. The best of the two for each band in terms of mean absolute error is selected for coding. In [61], the image is divided into non-overlapping regions, which allows parallel application of the methods described in [59] and [60].

B. Overview of the new standard

The CCSDS 123.0-B-2 standard is based on the FLEX compressor [62]. In turn, FLEX is based on the FL compressor [63], which was formalized as CCSDS 123.0-B-1. FLEX improves upon FL by adding adjustable lossy compression capabilities, while maintaining the option to perform lossless compression. The latest CCSDS compression standard extends FLEX by adding new features such as relative error limits, periodic error limit updating, and new prediction modes to facilitate hardware implementations. Very importantly, Issue 2 has been designed to retain many of FL's desirable properties, including low computational complexity, single-pass compression and decompression, automatic adaptation to the data being compressed and the ability to operate requiring a constant, reasonably sized memory space. Moreover, Issue 2 inherits all capabilities of CCSDS 123.0-B-1, allowing decompression of data output by the latter. These features make them suitable for use onboard spaceborne systems, including small satellite missions. Note that compressed images do not include synchronization markers or any other similar scheme. It is assumed that the transport layer will provide the ability to locate the next image in the event of a bit error or data loss.

The general structure of the Issue 2 compressor is shown in Fig. 1. Similar to CCSDS 123.0-B-1, input data —signed or unsigned integers— go through a predictor stage, in which previously coded information is employed to predict the value of the next sample to be compressed. As a main novelty of Issue 2, prediction errors are uniformly quantized. The quantization bin sizes are determined by the user's choice of absolute error limit, i.e., the maximum allowed absolute difference between original and reconstructed sample values, and/or the relative error limit, which controls the maximum ratio of the error to the sample's predicted value. Quantized data are then mapped to non-negative integers, which then are input to the entropy coder.

When non-zero error limits are selected, quantizer indices represent approximations of the aforementioned prediction errors, instead of the actual values. In this case, data output by the predictor stage typically exhibits lower entropy rates,

TABLE II: List of symbols referenced in this paper.

Symbol	Meaning
	General
$s_z(t)$	Original sample value (t-th sample of spectral band z).
D	Dynamic range in bits.
$s_{\min}$, $s_{\max}$	Minimum and maximum allowed sample values.
s_{mid}	Mid-range sample value.
N_X, N_Y	Horizontal and vertical spatial dimensions of the image.
N_Z	Number of spectral bands of the image.
	Sample representative calculation
$s''_z(t)$	Sample representative for $s_z(t)$.
Θ	Sample representative resolution.
ϕ_z	Sample representative damping for band z.
ψ_z	Sample representative offset for band z.
	Prediction
$\hat{s}_z(t)$	Predicted value for $s_z(t)$.
Ω	Prediction weight arithmetic resolution.
P	Number of previous bands to use for prediction.
$s_{z,y,x}$	Alternative notation for $s_z(t)$.
$\sigma_{z,y,x}$	Local sum for $s_{z,y,x}$.
$d_{z,y,x}$, $d^N_{z,y,x}$, $d^W_{z,y,x}$, $d^{NW}_{z,y,x}$	Local differences for $s_{z,y,x}$.
$\mathbf{U}_{z,y,x}$	Local difference vector for $s_{z,y,x}$.
$\mathbf{W}_{z,y,x}$	Prediction weight vector for $s_{z,y,x}$
$\hat{d}_{z,y,x}$	Predicted central local difference for $s_{z,y,x}$.
$\tilde{s}_z(t)$	Double resolution predicted value for $s_z(t)$.
$v_{\min}$, $v_{\max}$, t_{inc}, $\zeta_z^{(i)}$, ζ_z^*	User-specified weight update parameters.
	Quantization
$\Delta_z(t)$	Prediction error for $s_z(t)$.
$q_z(t)$	Quantizer index of $\Delta_z(t)$.
$s'_z(t)$	Clipped quantizer bin center for $\Delta_z(t)$.
a_z	Maximum absolute error in spectral band z.
r_z	Maximum relative error in spectral band z.
$m_z(t)$	Maximum reconstruction error $\lvert s_z(t) - s'_z(t)\rvert$.
	Quantizer index mapping
$\delta_z(t)$	Mapped quantizer index for $q_z(t)$.
$\theta_z(t)$	Scaled difference between $\hat{s}_z(t)$ and the closest of $s_{\min}$ and $s_{\max}$.
	Entropy coding
$U_{\max}$	Golomb-power-of-2 length limit.
$\Sigma_z(t)$	Accumulator value for $\delta_z(t)$.
$\Gamma(t)$	Counter value for $\delta_z(t)$.
γ^*	Sample-adaptive rescaling counter size.
$k_z(t)$	GPO2 code index for $\delta_z(t)$.
$\tilde{\Sigma}_z(t)$	High-resolution counter value for $\delta_z(t)$.
i	Hybrid code index.
T_i	Hybrid code entropy threshold constants.
L_i	Hybrid code symbol limit constants.
X	Hybrid code escape symbol.

which allows the coder to produce smaller compressed files. In order to make decompression possible, the decoder must be able to make the same predictions as the encoder. To guarantee this, when non-zero error limits are selected, prediction is done using so-called *sample representatives* instead of the original samples.

The rest of this section provides an informative description of the aforementioned functional blocks. For the sake of readability, some definitions in this description are simplified so as to not contemplate boundary cases, e.g., image edges when neighboring samples are involved. The interested reader is referred to [34] for complete, normative definitions. A list of the symbols employed hereinafter is available in Table II for ease of reference.

C. Predictor stage

The predictor stage is designed to process input samples sequentially in a single pass, producing one mapped quantizer index per input sample. While CCSDS 123.0-B-1 was designed to accept input samples of at most 16 bits, Issue 2 accepts bit depths, D, up to 32 bits. Hereinafter, $s_z(t)$ denotes the t-th sample of the z-th spectral band in raster scan order, and $\delta_z(t)$ its corresponding mapped quantizer index. To obtain $\delta_z(t)$, a prediction of the sample's original value —denoted as $\hat{s}_z(t)$— is computed as described in Section II-C3, and the prediction error is computed as

$$\Delta_z(t) = s_z(t) - \hat{s}_z(t). \tag{1}$$

This prediction error is then quantized as described in Section II-C1, to produce a quantizer index $q_z(t)$. This index is mapped to a non-negative value $\delta_z(t)$ —the output of the predictor stage— as described in Section II-C4.

The quantizer index is also transformed into its corresponding sample representative $s''_z(t)$, as described in Section II-C2. These representatives are then used to obtain the predicted values $\hat{s}_z(t)$ used in Eq. (1). As mentioned above, sample value prediction must be based on $s''_z(t)$ instead of $s_z(t)$ to avoid compressor-decompressor prediction differences when compression is not lossless.

1) Quantization: The CCSDS 123.0-B-2 standard allows quantization of each prediction error $\Delta_z(t)$ into a quantizer index $q_z(t)$ so that $\Delta_z(t)$ —and thus, also the input sample $s_z(t)$— can be reconstructed with maximum error $m_z(t)$. A quantizer with uniform bin size $2m_z(t)+1$ is used, i.e.,

$$q_z(t) = \text{sgn}(\Delta_z(t)) \cdot \left\lfloor \frac{|\Delta_z(t)| + m_z(t)}{2m_z(t)+1} \right\rfloor, \tag{2}$$

where the sgn function is defined as

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}. \tag{3}$$

Users control $m_z(t)$ indirectly by selecting the maximum absolute error a_z, the maximum relative error r_z, or both, for each spectral band z. When only absolute error limits are specified,

$$m_z(t) = a_z. \tag{4}$$

When only relative error limits are set,

$$m_z(t) = \left\lfloor \frac{r_z |\hat{s}_z(t)|}{2^D} \right\rfloor, \quad (5)$$

where $\hat{s}_z(t)$ is the predicted value for the original sample $s_z(t)$. Setting relative error limits allows reconstructing different samples with different degrees of precision. More specifically, samples *predicted* to have smaller magnitude are reconstructed with lower error. Note that predicted instead of actual sample values are used in Eq. (5) to keep the encoder and the decoder synchronized. Thus, absolute error bounds are not guaranteed when only a relative error limit $r_z > 0$ is specified. When both absolute and relative error limits are used, $m_z(t)$ is set to the minimum of equations (4) and (5). When lossless compression is desired in band z, users may set $a_z = 0$ or $r_z = 0$ so that $m_z(t) = 0$. This guarantees that $q_z(t) = \Delta_z(t)$, i.e., the original samples can be reconstructed exactly.

It is worth emphasizing that error limits can be set individually for each spectral band. With this mechanism, higher-importance bands can be reconstructed with greater fidelity (even perfect fidelity), whilst lesser-priority bands can be represented with lower fidelity using smaller compressed data volumes [56], [64]–[66]. Furthermore, the *periodic error limit update* option can be activated so that different fidelity choices can be adapted within a band. This option is useful to meet a given downlink transmission rate constraint, and/or to better preserve image regions expected to contain features of interest. It should be highlighted that the standard does not define a specific method for selecting error limit values, e.g., to meet a given downlink rate. This is because error limit values are encoded in the bitstream, and thus the decoder does not need to know how those error limits were selected.

2) Sample representatives: The decompressor must duplicate the prediction operation performed by the compressor, but in general the original image samples $s_z(t)$ cannot be perfectly reconstructed from the compressed bitstream because of information lost during the quantization stage. Consequently, the prediction calculation (in both the compressor and decompressor) is performed using *sample representatives* $s''_z(t)$ in place of the original samples $s_z(t)$.

A naive solution to this problem is to use the central point $s'_z(t)$ of the quantizer bin whose index $q_z(t)$ is transmitted to the decoder. The quantizer bin center $s'_z(t)$ can be calculated as

$$s'_z(t) = \text{clip}\left(\hat{s}_z(t) + q_z(t) \cdot (2m_z(t) + 1),\, s_{\text{min}},\, s_{\text{max}}\right), \quad (6)$$

where s_{min} and s_{max} are respectively the minimum and maximum values allowed for an input sample, and

$$\text{clip}(x,\, a,\, b) = \min(b,\, \max(a,\, x)) \quad (7)$$

guarantees that $s'_z(t)$ falls within the allowed value range. However, using the quantizer bin center $s'_z(t)$ as the sample representative $s''_z(t)$ for prediction does not always minimize compressed data volume [37]. This is true even for $m_z(t) = 0$, i.e., lossless compression.

In the CCSDS 123.0-B-2 standard, three user-specified parameters can be used to adjust the choice of $s''_z(t)$. These are the sample representative resolution (Θ), damping (ϕ_z), and offset (ψ_z) parameters. Based on them, sample representatives $s''_z(t)$ are defined as an integer approximation to

$$\frac{\phi_z}{2^\Theta}\hat{s}_z(t) + \left(1 - \frac{\phi_z}{2^\Theta}\right)\left(s'_z(t) - \frac{\psi_z}{2^\Theta}\text{sgn}\left(q_z(t)\right) m_z(t)\right). \quad (8)$$

Regardless of the parameter choice, sample representatives always fall between $s'_z(t)$ and $\hat{s}_z(t)$. Parameter Θ determines the precision with which representatives are computed. Parameter ϕ_z limits the effect of noisy samples in the representative calculation. In turn, parameter ψ_z establishes a bias towards $s'_z(t)$ or $\hat{s}_z(t)$, depending on its value. While Θ is defined for the whole image, ϕ_z and ψ_z can be chosen on a band-by-band basis. Setting $\phi_z = \psi_z = 0$ causes sample representatives to be equal to $s'_z(t)$, larger values of ϕ_z and/or ψ_z produce representatives closer to $\hat{s}_z(t)$. Note that, depending on the parameter choice, $s''_z(t)$ may not be contained in the quantizer bin identified by $q_z(t)$. Empirical results indicate that setting the damping and offset parameters to values different from zero tend to provide larger benefits to compression performance when spectral bands are closer in wavelength, and for images with larger noise prevalence [37].

3) Prediction: The predicted sample value $\hat{s}_z(t)$ for an input sample $s_z(t)$ is computed causally using sample representatives from spectral bands $z-P, \ldots, z$, where $P \geq 0$ is a user-defined parameter. Within each band, previous sample representatives are used to compute so-called *local sums*. These can be regarded as a preliminary, scaled estimates of the actual sample value. Local sums, in combination with the sample representatives, are used to compute *local differences*. The predicted value $\hat{s}_z(t)$ is then calculated using the local sum in the current band z, as well as a weighted sum of local differences from the current and previous bands. Local sums can be understood as a local mean subtraction, and prediction as being made in the mean-subtracted domain. Fig. 2 shows an overview of the prediction process. Its stages are more precisely described in what follows.

Local sums are computed from previous sample representatives using one of the four available modes. Similar to CCSDS 123.0-B-1, each mode is either *neighbor-oriented* or *column-oriented*. As a novelty of Issue 2, modes can now be *narrow* instead of *wide*. The sample representatives used to calculate the local sums depend on the selected mode, as depicted in Fig. 3. In the figure and hereinafter, $s_{z,y,x}$ is used to denote the current sample $s_z(t)$, which makes explicit the band index z, as well as the spatial coordinates (x, y) within the band. In all modes, the highlighted sample representatives are multiplied by the factor indicated in the aforementioned figure, and added together to obtain the local sum $\sigma_{z,y,x}$ corresponding to $s_{z,y,x}$. For instance, the narrow neighbor-oriented local sums are computed as

$$\sigma_{z,y,x} = s''_{z,y-1,x-1} + 2s''_{z,y-1,x} + s''_{z,y-1,x+1}. \quad (9)$$

As can be observed in Fig. 3, column-oriented local sums employ sample representatives at the same x coordinate, whereas neighbor-oriented sums also use sample representatives at contiguous x coordinates. In turn, the new narrow option removes dependency on $s''_{z,y,x-1}$, which facilitates pipelining

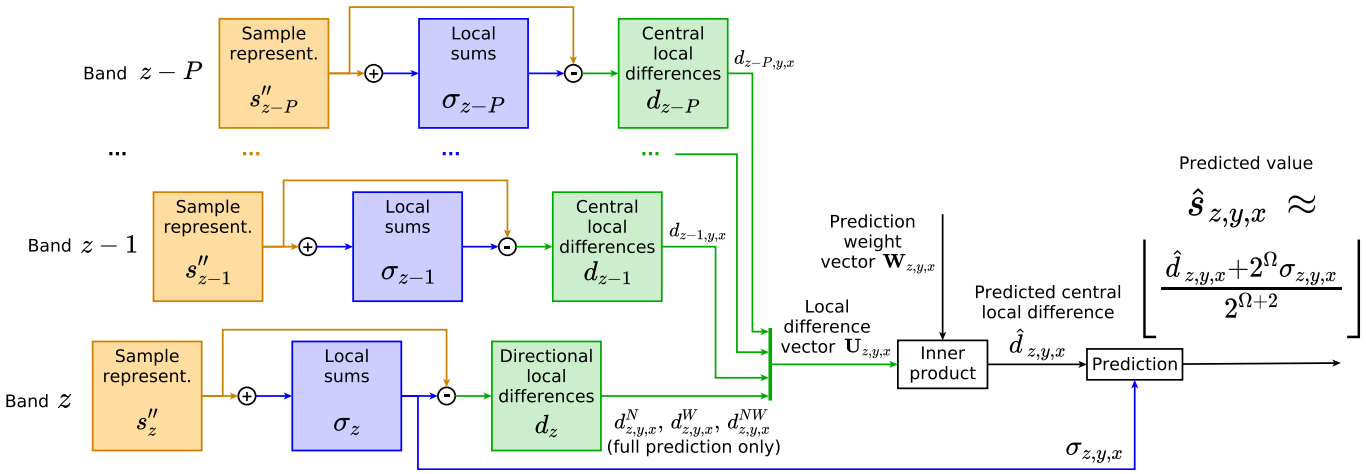
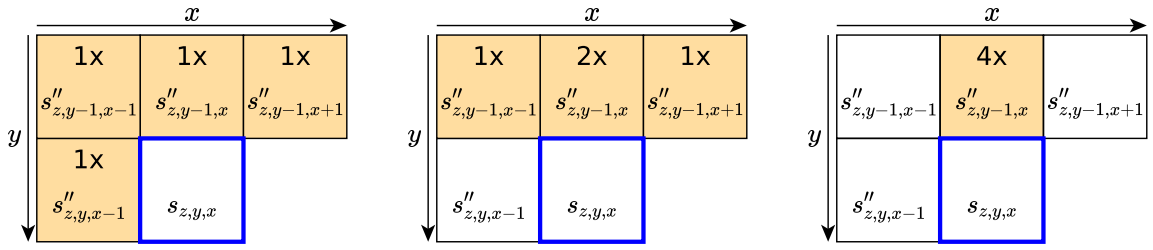


Fig. 2: Overview of the prediction block in CCSDS 123.0-B-2.



(a) Wide neighbor-oriented. (b) Narrow neighbor-oriented. (c) Column-oriented.

Fig. 3: Local sum calculation modes available in Issue 2. The current sample position is highlighted in blue. Sample representatives employed for the corresponding local sum are highlighted in orange.

in a hardware implementation, at the cost of some compression performance loss [37]. Note that wide and narrow column-oriented modes are identical in the general case. Notwithstanding, only the wide column-oriented mode uses $s''_{z,y,x-1}$ for calculating local sums at the first row —i.e., $y = 0$— of each spectral band.

Local differences are computed based on the sample representatives and the local sums. For an input sample $s_{z,y,z}$, up to four local difference types are computed: the central difference ($d_{z,y,x}$), and three directional differences, i.e., north ($d^N_{z,y,x}$), west ($d^W_{z,y,x}$), and north-west ($d^{NW}_{z,y,x}$). They are defined as follows:

$$\begin{aligned} d_{z,y,x} &= 4s''_{z,y,x} - \sigma_{z,y,x}, \\ d^N_{z,y,x} &= 4s''_{z,y-1,x} - \sigma_{z,y,x}, \\ d^W_{z,y,x} &= 4s''_{z,y,x-1} - \sigma_{z,y,x}, \\ d^{NW}_{z,y,x} &= 4s''_{z,y-1,x-1} - \sigma_{z,y,x}. \end{aligned} \tag{10}$$

The predicted sample value is then computed using either the so-called *full* or *reduced* prediction modes. In the *full* prediction mode, the *local difference vector* $\mathbf{U}_{z,y,x}$ is defined using directional differences from the current spectral band, and central differences from the previous bands:

$$\mathbf{U}_{z,y,x} = \left[d^N_{z,y,x},\, d^W_{z,y,x},\, d^{NW}_{z,y,x},\, d_{z-1,y,x},\, \ldots,\, d_{z-P,y,x}\right]. \tag{11}$$

In the *reduced* prediction mode, the local difference vector uses only central differences from previous bands:

$$\mathbf{U}_{z,y,x} = [d_{z-1,y,x},\, \ldots,\, d_{z-P,y,x}]. \tag{12}$$

In both modes, a *prediction weight vector* $\mathbf{W}_{z,y,x}$ is used to obtain a weighted sum of local differences called the *predicted central local difference* as

$$\hat{d}_{z,y,x} = \mathbf{W}^T_{z,y,x}\mathbf{U}_{z,y,x}. \tag{13}$$

The predicted sample is then calculated as an integer approximation to

$$\hat{s}_{z,y,x} \approx \left\lfloor \frac{\hat{d}_{z,y,x} + 2^{\Omega}\sigma_{z,y,x}}{2^{\Omega+2}} \right\rfloor, \tag{14}$$

where Ω is a parameter that controls arithmetic precision.

The initial prediction weight vector for each band, $\mathbf{W}_{z,0,0}$, can be defined based on default or user-provided values. In either case, vector elements are updated after processing each input sample $s_z(t)$. Updates are based on the obtained prediction error and several user-defined parameters —namely $v_{\min}$, $v_{\max}$, $\zeta^{(i)}_z$, ζ^*_z, and t_{inc}—, which control the rate at which weights are adapted to the original image statistics. More precisely, smaller values of $\zeta^{(i)}_z$, ζ^*_z, $v_{\min}$, $v_{\max}$, and $1/t_{\text{inc}}$ typically produce larger weight updates. This results in faster adaptation to source statistics, at the cost of worse steady-state compression performance [37].

It is important to highlight that the existence of two prediction modes (full and reduced), as well as two different local mean types (column-oriented and neighbor-oriented) is present in Issue 2 so that prediction is effective for image data produced by different types of instruments. For instance, when streaking artifacts are present in the images, reduced column-oriented prediction tends to produce the best results [37].

4) Quantizer index mapping: Prediction errors $\Delta_z(t)$ obtained in Eq. (1), as well as their corresponding quantizer indices $q_z(t)$ defined in Eq. (2), may be negative. However, entropy coders available in CCSDS 123.0-B are defined for non-negative input values. The quantizer index mapping stage depicted in Fig. 1 provides a one-to-one mapping between valid quantizer indices and non-negative values, referred to as *mapped quantizer indices*, and denoted $\delta_z(t)$.

This functional block remains unaltered in respect to the previous Issue of the standard [35]. A key property of this mapping is that indices can be represented using the same number of bits as the original image. This is true because predicted values are guaranteed to satisfy $\hat{s}_z(t) \in [s_{\min},\, s_{\max}]$, i.e., predictions do not exceed the range of allowed sample input values given bit depth D. Thus, the number of possible prediction errors equals the number of elements in the aforementioned interval. Based on this, the mapping is defined as

$$\delta_z(t) = \begin{cases} |q_z(t)| + \theta_z(t), & |q_z(t)| > \theta_z(t) \\ 2|q_z(t)|, & 0 \leq (-1)^{\tilde{s}_z(t)} q_z(t) \leq \theta_z(t) \\ 2|q_z(t)| - 1, & \text{otherwise} \end{cases} \quad (15)$$

where $\tilde{s}_z(t)$ is a double-resolution version of the predicted sample value defined in Section II-C3 and $\theta_z(t)$ is the difference between the predicted value and the nearest interval endpoint, i.e.,

$$\theta_z(t) = \min\left(\left\lfloor \frac{\hat{s}_z(t) - s_{\min} + m_z(t)}{2m_z(t)+1} \right\rfloor, \right. \quad (16)$$

$$\left. \left\lfloor \frac{s_{\max} - \hat{s}_z(t) + m_z(t)}{2m_z(t)+1} \right\rfloor \right). \quad (17)$$

D. Encoder stage

The encoder stage compresses the sequence of mapped quantizer indices $\delta_z(t)$ produced by the predictor stage into a variable-length bitstream. This operation is reversible, meaning that an identical sequence of mapped quantizer indices can be recovered from the bitstream. These indices allow an exact or an approximate reconstruction of the input image, depending on the error limits set in the predictor stage.

In Issue 2, three coders are available for this purpose: *sample-adaptive*, *block-adaptive* and *hybrid*. The user must select one of them to code all mapped quantizer indices for an image. The first two encoding options were already present in the previous Issue of the standard [35], while the hybrid coder is new in Issue 2. The hybrid coder tends to provide better compression performance than the other two options, but the benefit may be small when compression is lossless. An overview of the three available coders is provided in the rest of this section.

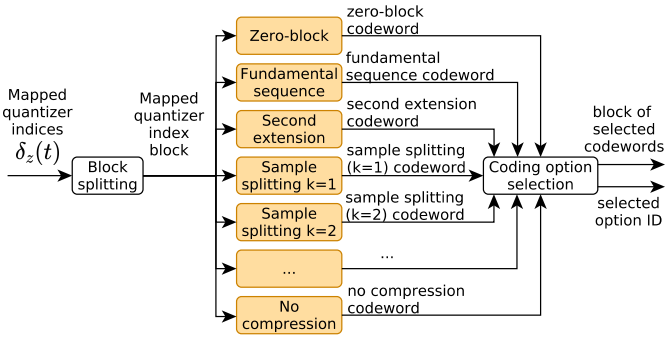


Fig. 4: Overview of the block-adaptive entropy coder. Coding options executed in parallel for each block are highlighted in orange.

1) Block-adaptive coder: The block-adaptive coder is a separate CCSDS standard, originally specified in [38] and later extended in [40], based on Rice coding. In this coder, samples are partitioned into disjoint blocks of fixed length of between 8 and 64 samples. Each block is encoded using the most effective of five available coding methods: zero-block, second extension, fundamental sequence, sample splitting, and no-compression. A simplified diagram of this process is shown in Fig. 4. The interested reader is referred to [67] for a summary of key operational concepts and a detailed performance analysis of this coder.

2) Sample-adaptive coder: In the sample-adaptive coder, each mapped quantizer index $\delta_z(t)$ is compressed using a variable-length codeword from a family of length-limited Golomb-power-of-2 (GPO2) codes. Each GPO2 code is identified by an index k, which is selected based on the statistics of previously coded samples. Given k and $\delta_z(t)$, the selected codeword is denoted as $\mathfrak{R}_k(\delta_z(t))$ and defined as follows:

(a) If $\lfloor \delta_z(t)/2^k \rfloor < U_{\max}$, $\mathfrak{R}_k(\delta_z(t))$ consists of $\lfloor \delta_z(t)/2^k \rfloor$ zeros, followed by a one, followed by the k least significant bits of the binary representation of $\delta_z(t)$.
(b) Otherwise, $\mathfrak{R}_k(\delta_z(t))$ consists of $U_{\max}$ zeros, followed by the binary representation of $\delta_z(t)$ using D bits.

Here, $U_{\max}$ is a user-specified parameter used to limit the maximum codeword length, and D is the image's bit depth.

Two variables are used to keep track of the input data statistics and choose the GPO2 family's index $k_z(t)$ to code $\delta_z(t)$: an *accumulator* $\Sigma_z(t)$ and a *counter* $\Gamma(t)$. The ratio of these two variables determines $k_z(t)$:

(a) If $2\Gamma(t) > \Sigma_z(t) + \lfloor 49\Gamma(t)/2^7 \rfloor$, then $k_z(t) = 0$.
(b) Otherwise, $k_z(t)$ is the largest positive integer such that

$$\begin{aligned} k_z(t) &\leq D - 2, \\ \Gamma(t)2^{k_z(t)} &\leq \Sigma_z(t) + \lfloor 49\Gamma(t)/2^7 \rfloor. \end{aligned} \quad (18)$$

Mapped quantizer indices typically follow a non-stationary geometric distribution, for which $k_z(t)$ is a good parameter estimator. Note that the counter and accumulator variables are initialized based on user-specified parameters.

The values of the counter and the accumulator variables are updated *after* coding each input sample $\delta_z(t-1)$. More specifically, Γ is increased by 1 and Σ is increased by $\delta_z(t-1)$. In addition, both Γ and Σ are periodically divided by 2 (rounding

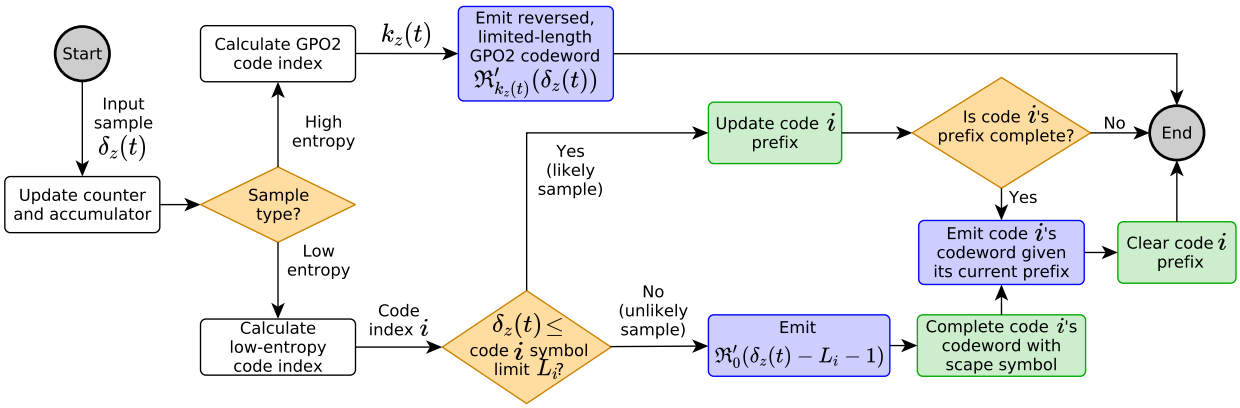


Fig. 5: Flow diagram of CCSDS 123.0-B-2's hybrid coder. Logical decisions are highlighted in orange. Processes that update the codes' internal state are highlighted in green. Processes that emit codewords are highlighted in blue.

down) to enable calculation using finite-precision arithmetic. This division is hereinafter referred to as *renormalization*.

3) Hybrid coder: The hybrid coder uses statistics of previously encoded data to classify each input mapped quantizer index as either a *high-entropy* or a *low-entropy* sample. High-entropy samples are compressed using a variation of the length-limited GPO2 code family described in Section II-D2. Low-entropy samples are coded using another family of 16 variable-to-variable length codes, i.e., several input samples can be encoded with a single codeword. A detailed description of these variable-to-variable length codes can be found in [68]. The ability to adaptively switch between GPO2 and variable-to-variable length codes gives this code the name *hybrid*.

Variable-to-variable length codes enable very efficient compression of highly predictable (low-entropy) samples, which become more prevalent when near-lossless error limits are used in the predictor stage. Meanwhile, variable-to-variable length codes introduce variability in the latency between the arrival of a low-entropy mapped quantizer index and the output of a codeword that encodes it. To accommodate this, codewords emitted by the hybrid coder are designed so that they can be decoded in reverse order. This is possible thanks to two main properties of the coder. First, output codewords are suffix-free rather than prefix-free. Second, the compressed image ends with a specification of the final state of the coder. A set of so-called *flush* tables is provided in the standard to signal the code states in an unambiguous an compact manner. Reverse decoding allows simpler and more memory-efficient implementations than FLEX's original hybrid entropy coder [62]. The rest of this section describes Issue 2's hybrid coder. A flow diagram of this coder's logic is provided in Fig. 5 to support this description.

Classification of samples as high or low entropy is performed using a similar statistical approach to that of the sample-adaptive coder. Two variables are used to keep track of these statistics: a counter $\Gamma(t)$ and a *high-resolution accumulator* $\tilde{\Sigma}_z(t)$. These variables are updated as in the sample-adaptive coder with two main differences. First, variables are updated *before* coding the input sample. This is done so that decoding can proceed in reverse order. To this effect, the least significant bit of the accumulator variable is output before renormalization so that the decoder can invert this process. Second, $\tilde{\Sigma}_z(t)$ is increased by $4\delta_z(t)$ instead of $\delta_z(t)$ to enable a more precise estimation of the input data statistics. The ratio $\tilde{\Sigma}_z(t)/\Gamma(t)$ determines whether a sample is a high-entropy or low-entropy symbol. More specifically, $\delta_z(t)$ is defined to be high entropy if and only if

$$\tilde{\Sigma}_z(t) \cdot 2^{14} \geq T_0 \cdot \Gamma(t), \tag{19}$$

where T_0 is a constant provided in the standard. This definition allows image regions that are well predicted to be coded with low-entropy codes, while using the high-entropy mode otherwise.

Each high-entropy sample is encoded using a family of reversed length-limited GPO2 codes. As in the sample-adaptive case, each code is identified by an index $k_z(t)$. For the hybrid coder, $k_z(t)$ is the largest positive integer that satisfies

$$\begin{aligned} k_z(t) &\leq \max(D-2, 2), \\ \Gamma(t)2^{k_z(t)+2} &\leq \tilde{\Sigma}_z(t) + \left\lfloor 49\Gamma(t)/2^5 \right\rfloor. \end{aligned} \tag{20}$$

The codeword emitted for the high-entropy sample $\delta_z(t)$, $\mathfrak{R}'_{k_z(t)}(\delta_z(t))$ is defined as follows:

(a) If $\left\lfloor \delta_z(t)/2^{k_z(t)} \right\rfloor < U_{\max}$, then $\mathfrak{R}'_{k_z(t)}(\delta_z(t))$ consists of the $k_z(t)$ least significant bits of the binary representation of $\delta_z(t)$, followed by a one, followed by $\left\lfloor \delta_z(t)/2^{k_z(t)} \right\rfloor$ zeros.

(b) Otherwise, $\mathfrak{R}'_{k_z(t)}(\delta_z(t))$ consists of the D-bit binary representation of $\delta_z(t)$ followed by $U_{\max}$ zeros.

Low-entropy samples are processed with one of 16 available variable-to-variable length codes. The code index used to process a low-entropy sample $\delta_z(t)$ is the largest i satisfying

$$\tilde{\Sigma}_z(t) \cdot 2^{14} < \Gamma(t) \cdot T_i, \quad 0 \leq i \leq 15, \tag{21}$$

where $T_0, \ldots, T_{15}$ are constants provided in the standard, and T_0 is used in Eq. (19). This definition allows the magnitude of recent prediction errors to determine the next variable-to-variable length code to be used.

Each code i has a prefix of previously input samples. When a sample is processed, a symbol is added to the corresponding code's prefix. The standard defines a list of *complete* prefixes for each code. At this point, if code i's prefix matches any of

TABLE III: Summary of employed corpus properties. Entropy (in bits) is averaged for all images in each row.

Instrument	Short name	Image Type	Bit depth D	Entropy	#Bands	Width	Height	#Images
Atmospheric Infrared Sounder	AIRS	raw	12	11.2	1501	90	135	1
Airborne Visible/Infrared Imaging Spectrometer	AVIRIS	raw	15	12.6	224	680	512	1
		raw	10	8.6	224	614	512	1
		calibrated	13	10.3	224	677	512	13
Compact Airborne Spectrographic Imager	CASI	raw	12, 13, 15	11.6	72	406	1225	3
Compact Reconnaissance Imaging Spectrometer for Mars	CRISM	FRT, raw	11	10.1	107	640	510	2
		FRT, raw	12, 13	10.4	438	640	510	2
		FRT, raw	12, 13	10.6	545	640	510	2
		HRL, raw	12, 13	11.2	545	320	450	2
		MSP, raw	11	9.8	74	64	2700	2
Hyperion	Hyperion	raw	12	8.5	242	256	1024	3
Infrared Atmospheric Sounding Interferometer	IASI	calibrated	12	11.0	8461	66	60	1
Landsat	Landsat	raw	8	6.6	6	1024	1024	3
Moon Mineralogy Mapper	M3	target, raw	12	9.7	260	640	512	2
		global, raw	11, 12	9.4	86	320	512	2
Moderate Resolution Imaging Spectroradiometer	MODIS	night, raw	12	10.8	17	1354	2030	2
		day, raw	12, 13	8.6	14	1354	2030	2
		500 m, raw	12, 13	11.1	5	2708	4060	2
		250 m, raw	12	10.4	2	5416	8120	2
Meteosat Second Generation	MSG	calibrated	10	8.2	11	3712	3712	1
PLEIADES High Resolution	PLEIADES	HR, simulated	12	10.8	4	224	2465	1
		HR, simulated	12	10.2	4	224	2448	3
SWIR Full Spectrum Imager	SFSI	calibrated	15	9.9	240	452	140	1
		raw	9, 11	7.4	240	496	140	2
Système Pour l'Observation de la Terre 5 High Resolution Geometric	SPOT5	HRG, processed	8	6.8	3	1024	1024	1
Vegetation	Vegetation	raw	10	9.4	4	1728	10080	2

those complete prefixes, a codeword is emitted that uniquely identifies that prefix and its associated sequence of input samples. After that, the prefix for that code is cleared.

It is worth noting that complete prefixes defined for code i cannot contain sample values satisfying $\delta_z(t) > L_i$, where $L_0, \ldots, L_{15}$ are constants defined in the standard. When such a sample is processed —i.e., referred to as an *unlikely* sample—, $\mathfrak{R}'_0(\delta_z(t) - L_i - 1)$ is emitted and an escape symbol X is added to the prefix instead of $\delta_z(t)$. Adding X to any code's prefix is guaranteed to make it complete, and trigger emission of an output codeword. The input symbol limit L_i limits the size of the input alphabet in the low-entropy codes, by treating all unlikely symbols in the same way. This allows us to reduce the number of codewords in a code. Since escape symbols occur with low probability, the efficiency with which these residual values are encoded has only a small impact on overall coding effectiveness.

III. Compression performance

A. Experimental setup

The lossless and near-lossless compression performance of Issue 2 is illustrated in this section. Results are provided for the block-adaptive and sample-adaptive entropy coders, already present in Issue 1, and compared to those of the new hybrid coder defined in Issue 2. Its computational complexity is comprehensively addressed in [69], so execution time results are not presented here. Empirical results have been obtained using a very varied corpus of 17 multispectral images, 38 hyperspectral images, and 2 sounder data samples. These were generated by 14 different instruments deployed in real missions, except for the PLEIADES images, which are simulated. Most of the images included are *raw*, giving more weight to the direct compression of images as they are acquired, while non-raw instances that were processed after acquisition are also included to represent some possible on-board calibration. Both pushbroom and whiskbroom sensors are covered in the corpus, and include the streaking artifacts that characteristic of pushbroom instruments (such as Hyperion) in uncalibrated images. A varied range of spectral separations are considered, and examples of images with significant noise levels (M3) or that are acquired with airborne instruments (CASI) are included as well. Regarding dynamic range, all hyperspectral and sounding instruments produce data with bit depths of at least 11 bits, whereas for multispectral instruments samples of lower bit depths are available too.

A summary of this corpus, which was produced by the CCSDS' MHDC working group, is provided in Table III. All images are publicly available, except for those produced by the IASI and MSG instruments due to licensing restrictions.[2] The *Entropy* column in the table represents the zero-order

[2]Download links for the test images can be found at at http://cwe.ccsds.org/sls/docs/sls-dc/123.0-B-Info/TestData.

entropy of the images. Note that this is not a strict bound on compression efficiency, and should only be regarded as an assessment of the difficulty of compressing the images.

Performance results are obtained by invoking Issue 2's compressor with the default set of parameters described in [37], except for the Hyperion, IASI, MODIS, and SPOT5 instruments. For these, the following parameters are modified to enhance compression performance: $t_{\text{inc}} = 2^9$, $v_{\text{min}} = v_{\text{max}} = 0$, $U_{\text{max}} = 32$, $\gamma^\star = 11$ and $\gamma_0 = 4$. Full prediction with wide, neighbor-oriented local sums is used in most images, including the four aforementioned instruments. Column-oriented local sums are employed for images that present streaking artifacts, i.e., when average sample values exhibit strong differences for contiguous x positions. A full analysis of the impact on performance of parameter tuning, as well as an identification of images with streaking artifacts, can be found in [37].

To provide a comparison baseline, the authors' implementation of CCSDS 122.1-B-1, the reference implementation of the JPEG-LS standard[3] and the original authors' implementation of M-CALIC [70] have been included in the comparison as well. For CCSDS 122.1-B-1, the best-performing configuration in terms of rate-distortion, i.e., float DWT and spectral POT, is used. JPEG-LS is arguably the best-known compression standard that offers low complexity and supports both lossless and near-lossless regimes. In turn, M-CALIC is another low-complexity algorithm well-known for its competitive compression performance. Note that, since JPEG-LS does not admit an arbitrary number of spectral bands, images are reshaped by concatenating the bands along the Y axis. More specifically, an image with width, height and number of bands respectively equal to N_X, N_Y and N_Z is transformed into a one-band image with the same width and height equal to $N_Y \cdot N_Z$. No attempt is made to perform decorrelation across spectral bands for JPEG-LS. In contrast, M-CALIC is designed specifically to exploit spectral redundancy in hyperspectral images.

B. Lossless compression results

Lossless compression results are obtained for all Issue 2's entropy coders, for JPEG-LS, and for M-CALIC by setting the absolute error limit to zero. For each image I in the test corpus, the compression ratio is defined as

$$\text{CR}(I) = \frac{N_X \cdot N_Y \cdot N_Z \cdot D}{\text{compressed data size (bits)}}. \tag{22}$$

Based on this definition, higher compression ratio values indicate better compression. A distribution of the obtained compression ratios for each compressor is shown in Fig. 6. Vertical bar heights indicate the relative frequency of each range of compression ratios. The average compression ratio, plus/minus one standard deviation is denoted with a dot and two horizontal bars. Note that aggregated results presented here and in Section III-C are not necessarily representative of any particular image or instrument. This is due to their different statistical properties, and the fact that different number of images are available for each instrument.

[3]The employed JPEG-LS implementation is available at https://github.com/thorfdbg/libjpeg. To attain lossless and near-lossless compression, this compressor is invoked with parameter `-ls 0`.

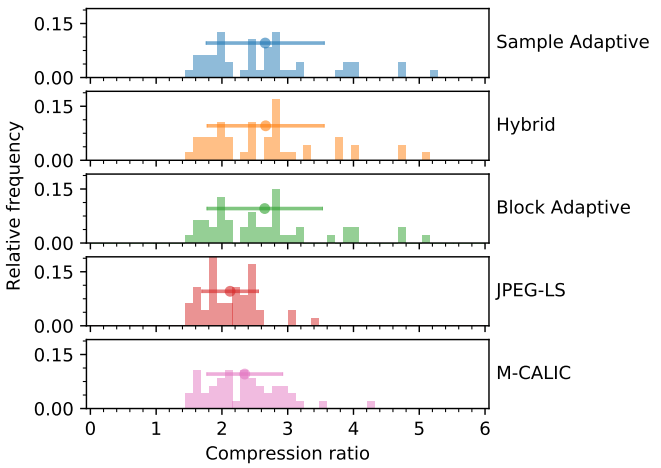


Fig. 6: Distribution of lossless compression ratios.

As can be observed, all three entropy coders in Issue 2 yield similar compression ratio distributions and average values. In turn, JPEG-LS and M-CALIC produce average compression ratios respectively 25% and 13% lower than those of Issue 2. These differences can be explained by a more advanced predictor stage used in Issue 2.

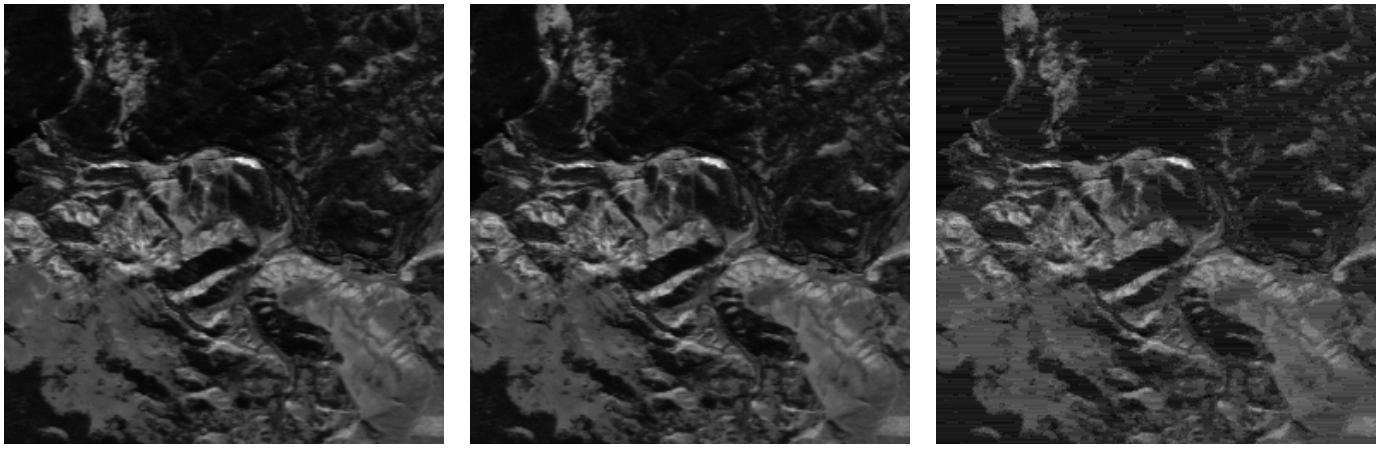
To provide further insight, average compression ratios grouped by instrument are shown in the "Lossless" columns of Table IV for Issue 2 using the hybrid coder, for JPEG-LS, and for M-CALIC. Consistent with the previous discussion, the CCSDS compressor yields higher compression efficiency than JPEG-LS and M-CALIC for most instruments. Improvements of up to 63.7% and 63.4%, respectively, can be observed. Only for the MODIS instrument does JPEG-LS perform better, yielding an average compression ratio 7.7% higher than Issue 2's with the hybrid coder. In turn, M-CALIC improves upon JPEG-LS in all cases, and is able to yield results between 0.3% and 8.9% better than Issue 2 for five of the tested instruments. These differences can be explained by the fact that M-CALIC employs an arithmetic entropy coder, which enables better modeling of the source's statistics, although at the cost of higher computational complexity.

C. Near-lossless compression results

Near-lossless compression results are obtained for all three entropy coders in CCSDS 123.0-B-2, as well as for JPEG-LS and M-CALIC, by limiting the maximum absolute error in any pixel of the reconstructed images. This error is hereinafter denoted as peak absolute error (PAE). Two illustrative examples of near-lossless compression using Issue 2 and JPEG-LS are provided in Fig. 7. In the top row, it can be observed that Issue 2's hybrid coder enables higher image quality –i.e., lower PAE– at similar albeit smaller compressed data sizes. Furthermore, for sufficiently low PAEs, reconstructed images are hardly distinguishable from the originals. In turn, the bottom row illustrates how moderately larger PAEs introduce

TABLE IV: Average compression ratios results grouped by instrument.

Instrument	CCSDS 123.0-B-2 (Hybrid coder)					JPEG-LS					M-CALIC				
	Lossless	PAE 1	PAE 2	PAE 5	PAE 16	Lossless	PAE 1	PAE 2	PAE 5	PAE 16	Lossless	PAE 1	PAE 2	PAE 5	PAE 16
AIRS	2.86	4.56	6.09	10.76	35.74	1.89	2.51	2.95	3.97	6.68	2.87	4.51	5.92	9.95	27.51
AVIRIS	3.11	5.28	7.66	15.29	37.52	1.90	2.56	3.03	4.09	7.06	3.01	4.82	6.31	10.23	21.36
CASI	2.29	3.22	3.96	5.91	12.21	1.66	2.08	2.36	2.96	4.38	2.27	3.17	3.87	5.63	11.09
CRISM	3.10	5.05	6.87	11.15	22.93	2.20	3.08	3.71	5.14	8.17	2.21	3.21	4.01	6.08	13.43
Hyperion	2.86	4.57	6.09	10.80	44.75	2.44	3.56	4.48	6.76	13.22	2.79	4.36	5.72	9.59	28.38
IASI	2.53	3.75	4.70	7.17	14.96	1.92	2.56	3.01	4.05	7.12	2.48	3.64	4.55	6.94	15.75
Landsat	2.35	4.12	6.24	12.80	41.88	2.13	3.68	5.09	8.46	20.33	2.37	3.97	5.40	9.25	19.51
M3	4.38	7.44	9.61	14.27	24.28	2.72	4.15	5.29	7.27	10.33	2.68	4.17	5.42	8.86	22.49
MODIS	1.94	2.60	3.07	4.12	7.35	2.09	2.77	3.24	4.27	6.95	2.13	2.72	3.22	4.35	7.39
MSG	2.77	4.49	6.06	10.01	24.18	2.64	4.20	5.39	8.08	14.78	2.73	4.12	5.31	8.22	17.45
PLEIADES	1.66	2.12	2.43	3.11	5.04	1.62	2.06	2.36	3.01	4.64	1.68	2.16	2.49	3.23	5.18
SFSI	3.07	5.18	7.02	11.97	53.21	2.58	3.75	4.65	6.99	16.50	2.91	4.39	5.65	9.13	30.05
SPOT5	1.55	2.21	2.74	4.22	10.00	1.45	2.03	2.48	3.63	6.69	1.54	2.22	2.74	4.07	8.90
Vegetation	1.95	2.77	3.40	5.04	10.54	1.87	2.61	3.16	4.42	7.78	2.03	2.86	3.51	5.08	10.05
All	2.67	4.20	5.55	9.07	22.98	2.12	3.00	3.67	5.17	9.20	2.35	3.44	4.34	6.70	15.43



(a) (b) (c)

(d) (e) (f)

Fig. 7: (a) Crop (256×256) of band 220 of original AVIRIS *f060925t01p00r12_sc00* image (calibrated, 16 bit); (b) and (c) co-located crops of the same AVIRIS image after reconstruction with CCSDS 123.0-B-2's hybrid coder (compressed at 2.4 bps) and JPEG-LS (2.9 bps) with absolute error limits 2 and 16, respectively; (d) Crop (128×128) of original SPOT5 *toulouse_spot5_xs_extract1* image (processed, 8 bit); (e) and (f) co-located crops of the same SPOT5 image after reconstruction with CCSDS 123.0-B-2's hybrid coder (1.1 bps) and JPEG-LS (1.4 bps) with absolute error limit 12. Brightness and magnification have been adjusted in all images to facilitate comparison. The SPOT5 images are presented using false color.

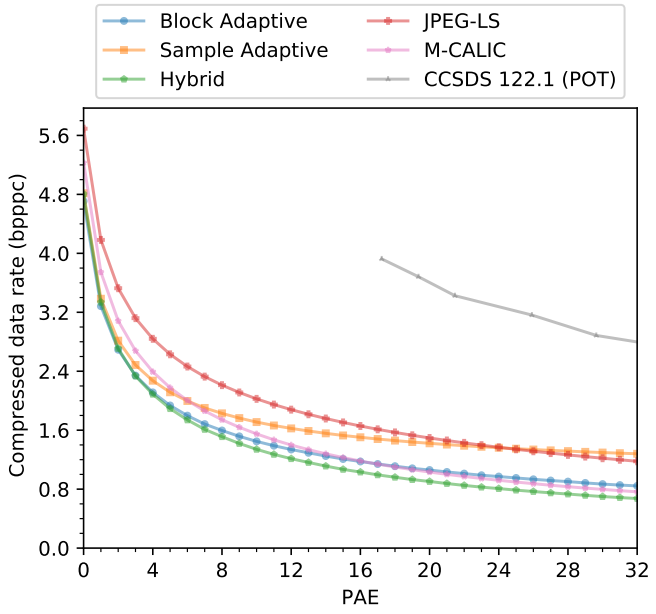


Fig. 8: Average compressed data rate in bits per sample as a function of the maximum absolute error.

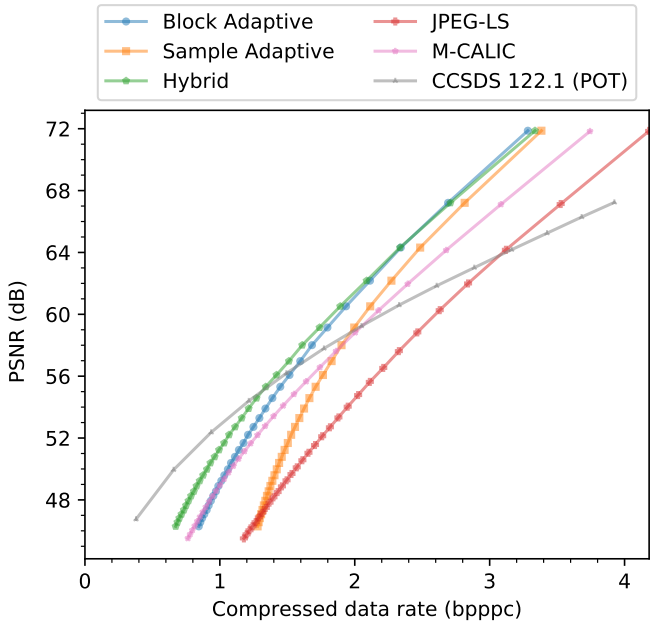


Fig. 9: Average PSNR results as a function of the average compressed data rate.

some texture artifacts but retain the image's structure, and might not hinder analysis tasks performed on it [30], [31]. Visual inspection of this row also reveals that Issue 2 introduces distortion patterns similar to those of JPEG-LS. This is as expected, since both algorithms apply quantization after prediction. It is worth noting that the choice of entropy coder in CCSDS 123.0-B does not affect the obtained reconstructed image, only the compressed data size. Compressed data rate differences aside, a similar discussion regarding visual quality applies for M-CALIC too. Notwithstanding, it is omitted here in the interest of space.

The rest of this section provides quantitative discussion of the compression performance of the aforementioned algorithms, in relation to the fidelity of the reconstructed images. For each compressor, PAE, and input image I, the *compressed data rate* expressed in bits per sample (bps) is computed as

$$\text{compressed data rate} = \frac{\text{compressed data size (bits)}}{N_X \cdot N_Y \cdot N_Z}. \quad (23)$$

In turn, the peak signal-to-noise ratio (PSNR) between I and its reconstructed counterpart $\hat{I}$ is defined as

$$\text{PSNR}(I, \hat{I}) = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}(I, \hat{I})}\right) \quad \text{(dB)}. \quad (24)$$

Here, MAX_I denotes the dynamic range of an image —i.e., $2^D - 1$, where D is I's bit depth—, and $\text{MSE}(I, \hat{I})$ is the mean squared error between I and $\hat{I}$, i.e.,

$$\text{MSE}(I, \hat{I}) = \frac{\sum_{x}^{N_X}\sum_{y}^{N_Y}\sum_{z}^{N_Z}\left(I_{z,y,x} - \hat{I}_{z,y,x}\right)^2}{N_X \cdot N_Y \cdot N_Z}. \quad (25)$$

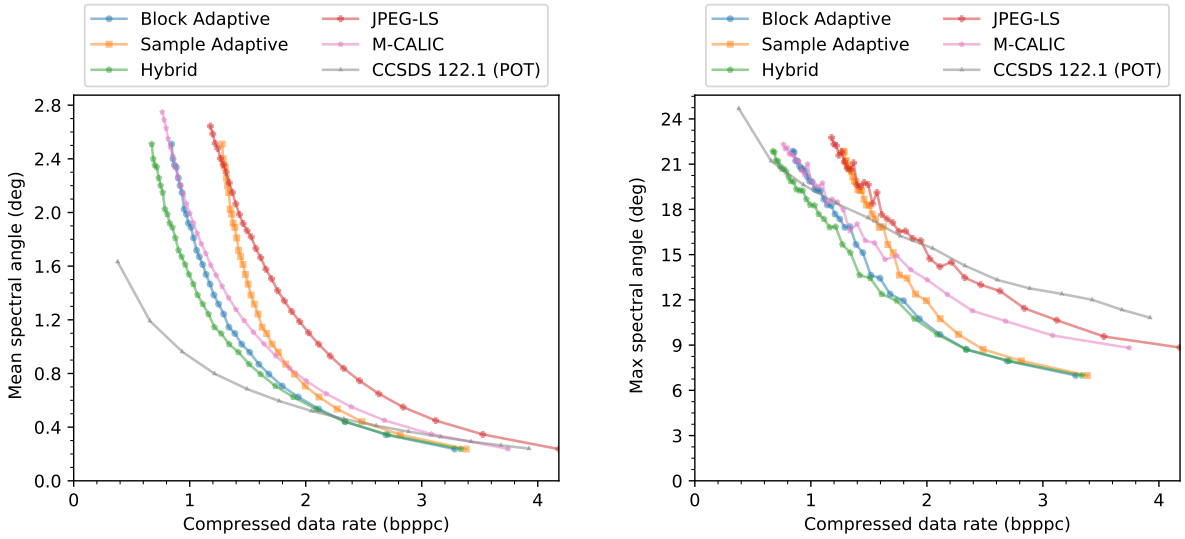
The spectral angle is computed at each (x, y) position for each original and reconstructed image pair, defined in [71] as

$$\alpha(x, y) = \cos^{-1}\left(\frac{\sum_z^{N_z} I_{z,y,x} \cdot \hat{I}_{z,y,x}}{\sqrt{\sum_z^{N_z} I_{z,y,x}^2} \cdot \sqrt{\sum_z^{N_z} \hat{I}_{z,y,x}^2}}\right). \quad (26)$$

The mean spectral angle and maximum spectral angle metrics are defined, respectively, as the average and maximum spectral angle for all (x, y) positions in the image.

Fig. 8 provides near-lossless compressed data rate results for the three entropy coders of Issue 2, for JPEG-LS, and for M-CALIC, setting PAE limits between 0 (lossless) and 32. For each coder and PAE value, the plotted value is the mean compressed data rate for all images in the corpus. Markers have been included in the figure at the integer PAE values for which data have been obtained, and linear interpolation is used between them for the sake of readability. Results indicate that for larger PAE values, differences between Issue 2's coders become more apparent than for the lossless case. When compared to the block-adaptive and sample-adaptive coders, the hybrid coder yields compressed data rates up to 0.2 bps and 0.6 bps better, respectively. For PAE values up to 5, both JPEG-LS and M-CALIC are outperformed by all entropy coders of Issue 2. For PAE value from 20 onwards, M-CALIC improves upon the block-adaptive coder. For PAEs larger than 25, JPEG-LS produces results better than the sample-adaptive coder. Notwithstanding, for PAE values of 2 and above, the hybrid coder's average results are consistently better than all other compressors for all tested PAE values.

The global results presented in Fig. 8 are complemented by Table IV, reporting also average compression ratios for several PAE values. In it, average compression ratios for each instrument are provided. It can be observed that per-instrument



(a) (b)

Fig. 10: Spectral angle metrics as a function of the compressed data rate: (a) mean spectral angle; (b) maximum spectral angle.

results are generally consistent with global averages, with similar exceptions as for the lossless case.

These behaviors are explained by the different predictor stages, and by the way in which each coder handles the low-entropy data prevalent in near-lossless compression. The sample-adaptive coder does not have a mode in which multiple input symbols are compressed in a single codeword. Therefore, the minimum length of any sample-adaptive codeword sets a lower bound for the compression rates achievable by this coder. Both JPEG-LS and the block-adaptive coder have so-called *run-length* modes that allow coding of consecutive zeros in a single codeword. Thus, their compression performance is increased as the prevalence of such runs is increased. In turn, the 16 stateful codes featured by the hybrid coder enable a more efficient processing of low-entropy data, including inputs that are not sequences of only zeros. Finally, M-CALIC's performance improvement for higher PAEs is due to its arithmetic entropy coder, which is close to optimal for most data distributions.

In addition to considering the compressed data rates and PAE of the reconstructed images, it is useful to consider other distortion metrics to better understand the efficiency of each coder. To complete the rate-distortion compression performance comparison, the average PSNR as a function of the average compressed data rate is plotted in Fig. 9. The mean spectral angle and maximum spectral angle metrics are plotted in Fig. 10a and Fig.**??**, respectively. All metrics are computed for each coder, PAE value (or target bitrate, for CCSDS 122.1-B-1), and test image, and the mean values are used in the plots. Markers are placed at the obtained data points, and linear interpolation is used between them to enhance readability. As in the previous case, the hybrid coder yields better fidelity results than the other near-lossless coders for all metrics, especially at low compressed data rates. This and other differences between compressors are comparable to those shown in Fig. 8, for similar reasons as above. When compared to CCSDS 122.1-B-1, all near-lossless codecs yield significantly better PAE results. This is as expected, since the CCSDS 122.1-B-1 standard is not designed to bound the maximum introduced error, but instead to minimize MSE. At low bitrates, i.e. below 1.25 bps, CCSDS 122.1-B-1 yields the best PSNR results of all tested codecs. Again, this can be explained by the minimization goal of this standard. At higher bitrates, the Hybrid coder of Issue 2 produces the best PSNR results, which illustrates the competitive performance of CCSDS 123.0-B-2. When spectral angles are considered, the relative performance of the near-lossless coders is very similar to the PAE and PSNR cases. In turn, for the mean spectral angle metric, CCSDS 122.1-B-1 improves upon all other coders for bitrates up to 2 bps. This can be explained by the fact that CCSDS 122.1-B-1 applies a spectral transform across all bands, instead of predicting pixel values using a local spatial and spectral neighborhood. Interestingly, when the maximum spectral angle is considered, CCSDS 123.0-B-2 yields better results than CCSDS 122.1-B-1 except for low bitrates, below 0.75 bps. This can be explained by the fact that CCSDS 123.0-B-2 is near-lossless, i.e., it bounds the maximum error introduced in any pixel of the image.

IV. Conclusions

Multispectral and hyperspectral imaging has become an invaluable tool for many commercial, scientific and defense applications of remote sensing. With the advent of sensors with enhanced spatial and spectral resolution, data compression is paramount to maximize the amount of valuable information retrieved from spaceborne systems. In particular, near-lossless compression can significantly improve the effective capacity of transmission channels, while providing strict control of the distortion introduced in the images. Even if rate-control

strategies are possible, strong quality guarantees are prioritized over obtaining constant data rates in near-real-time transmission. The CCSDS 123.0-B-2 compression standard published by the Consultative Committee on Space Data Systems enables specification of absolute and/or relative error limits at the image or band level. This is achieved via uniform in-loop quantization of prediction errors, obtaining higher performance at the expense of a simpler implementation. Since the decompressor does not have access to the original image samples, sample representatives are used instead in the predictor stage. To fully exploit the lower entropy rates exhibited by quantized data, a new hybrid entropy coder is defined for Issue 2. This coder includes 16 variable-to-variable length codes, which are selected on a sample-by-sample basis depending on the statistics of previously coded information. One last improvement over CCSDS 123.0-B-1 is the definition of narrow local sums that facilitate the design of highly efficient hardware implementations. Experimental results with a comprehensive corpus of test images indicate that the new hybrid coder yields competitive compression performance results, measurably improving upon other coding modes of Issue 2 as well as upon the JPEG-LS compression standard and the M-CALIC algorithm. The standard obtains state-of-the-art performance in absolute or relative error measurements, while other approaches may provide better performance in terms of quadratic error at very low rates.

Regarding future developments related to this standard, it is unlikely that major changes are introduced soon.

REFERENCES

[1] M. Parente, J. Kerekes, and R. Heylen, "A Special Issue on Hyperspectral Imaging [From the Guest Editors]," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 6–7, Jun. 2019.

[2] E. J. Ientilucci and S. Adler-Golden, "Atmospheric Compensation of Hyperspectral Data: An Overview and Review of In-Scene and Physics-Based Approaches," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 31–50, June 2019.

[3] 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.

[4] M. Malyy, Z. Tekic, and A. Golkar, "What Drives Technology Innovation in New Space?: A Preliminary Analysis of Venture Capital Investments in Earth Observation Start-Ups," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 1, pp. 59–73, Mar. 2019.

[5] J. Theiler, A. Ziemann, S. Matteoli, and M. Diani, "Spectral Variability of Remotely Sensed Target Materials: Causes, Models, and Strategies for Mitigation and Robust Exploitation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 8–30, June 2019.

[6] Y. Zhong, X. Wang, Y. Xu, S. Wang, T. Jia, X. Hu, J. Zhao, L. Wei, and L. Zhang, "Mini-UAV-Borne Hyperspectral Remote Sensing: From Observation and Processing to Applications," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 4, pp. 46–62, Dec 2018.

[7] 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.

[8] W. Sun and Q. Du, "Hyperspectral Band Selection: A Review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 118–139, Jun. 2019.

[9] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, 2019.

[10] P. Duan, X. Kang, S. Li, P. Ghamisi, and J. A. Benediktsson, "Fusion of Multiple Edge-Preserving Operations for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10 336–10 349, 2019.

[11] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "DAEN: Deep Autoencoder Networks for Hyperspectral Unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4309–4321, 2019.

[12] Y. Chen, K. Zhu, L. Zhu, X. He, P. Ghamisi, and J. A. Benediktsson, "Automatic Design of Convolutional Neural Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7048–7066, 2019.

[13] J. M. Haut, J. A. Gallardo, M. E. Paoletti, G. Cavallaro, J. Plaza, A. Plaza, and M. Riedel, "Cloud Deep Networks for Hyperspectral Image Analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9832–9848, 2019.

[14] B. Tu, X. Zhang, X. Kang, J. Wang, and J. A. Benediktsson, "Spatial Density Peak Clustering for Hyperspectral Image Classification With Noisy Labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5085–5097, 2019.

[15] K. Bhardwaj, S. Patra, and L. Bruzzone, "Threshold-Free Attribute Profile for Classification of Hyperspectral Images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7731–7742, 2019.

[16] X. Lu, L. Dong, and Y. Yuan, "Subspace Clustering Constrained Sparse NMF for Hyperspectral Unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3007–3019, 2020.

[17] C. J. Della Porta, A. A. Bekit, B. H. Lampe, and C. Chang, "Hyperspectral Image Classification via Compressive Sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8290–8303, 2019.

[18] J. Nalepa, M. Myller, and M. Kawulok, "Validating Hyperspectral Image Segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1264–1268, 2019.

[19] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant Attribute Profiles: A Spatial-Frequency Joint Feature Extractor for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–18, 2020.

[20] EUMETSAT, "IASI Level 1: Product Guide," Tech. Rep. EUM/OPS-EPS/MAN/04/0032, Sep. 2019.

[21] K. Turpie, S. Veraverbeke, R. Wright, M. Anderson, and D. Quattrochi, "NASA 2014 The Hyperspectral Infrared Imager (HyspIRI) - Science Impact of Deploying Instruments on Separate Platforms," Jet Propulsion Lab, techreport JPL-Publ-14-13, Jul 2014. [Online]. Available: http://hdl.handle.net/2060/20160001776

[22] S.-E. Qian, *Optical satellite data compression and implementation*. SPIE, 2013.

[23] B. Huang, *Satellite data compression*. Springer Science & Business Media, 2011.

[24] K. Sayood, *Introduction to data compression*, 5th ed. Morgan Kaufmann, 2017.

[25] S. Álvarez-Cortés, J. Serra-Sagristà, J. Bartrina-Rapesta, and M. W. Marcellin, "Regression Wavelet Analysis for Near-Lossless Remote Sensing Data Compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 790–798, 2020.

[26] D. Valsesia and E. Magli, "High-Throughput Onboard Hyperspectral Image Compression With Ground-Based CNN Reconstruction," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9544–9553, Dec. 2019.

[27] M. Díaz, R. Guerra, P. Horstrand, E. Martel, S. López, J. F. López, and R. Sarmiento, "Real-Time Hyperspectral Image Compression Onto Embedded GPUs," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 2792–2809, 2019.

[28] S.-E. Qian, *Optical Satellite Signal Processing and Enhancement*. SPIE, 2013.

[29] Z. Chen, Y. Hu, and Y. Zhang, "Effects of Compression on Remote Sensing Image Classification Based on Fractal Analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4577–4590, Jul. 2019.

[30] J. García-Sobrino, A. J. Pinho, and J. Serra-Sagristà, "Competitive Segmentation Performance on Near-Lossless and Lossy Compressed Remote Sensing Images," *IEEE Geosci. Remote Sens. Lett.*, 2019.

[31] F. Garcia-Vilchez, J. Muñoz-Marí, M. Zortea, I. Blanes, V. González-Ruiz, G. Camps-Valls, A. Plaza, and J. Serra-Sagristà, "On the impact of lossy compression on hyperspectral image classification and unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 2, pp. 253–257, 2010.

[32] I. Blanes, E. Magli, and J. Serra-Sagrista, "A Tutorial on Image Compression for Optical Space Imaging Systems," *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, no. 3, pp. 8–26, Sep. 2014.

[33] A. D. George and C. M. Wilson, "Onboard processing with hybrid and reconfigurable computing on small satellites," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 458–470, 2018.

[34] *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression. Blue Book. Issue 2*, Consultative Committee for Space Data Systems (CCSDS)

Std. CCSDS 123.0-B-2, Feb. 2019. [Online]. Available: https://public.ccsds.org/Pubs/123x0b2c1.pdf

[35] *Lossless Multispectral & Hyperspectral Image Compression. Silver Book. Issue 1*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 123.0-B-1-S, May 2012. [Online]. Available: https://public.ccsds.org/Pubs/123x0b1ec1s.pdf

[36] A. Kiely, M. Klimesh, I. Blanes, J. Ligo, E. Magli, N. Aranki, M. Burl, R. Camarero, M. Cheng, S. Dolinar, D. Dolman, G. Flesch, H. Ghassemi, M. Gilbert, M. Hernández-Cabronero, D. Keymeulen, M. Le, H. Luong, C. McGuiness, G. Moury, T. Pham, M. Plintovic, F. Sala, L. Santos, A. Schaar, J. Serra-Sagristà, S. Shin, B. Sundlie, D. Valsesia, R. Vitulli, E. Wong, W. Wu, H. Xie, and H. Zhou, "The new CCSDS Standard for Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression," in *6th International Workshop on On-Board Payload Data Compression (OBPDC)*, 2018.

[37] I. Blanes, A. Kiely, M. Hernández-Cabronero, and J. Serra-Sagristà, "Performance Impact of Parameter Tuning on the CCSDS-123.0-B-2 Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression Standard," *MDPI Remote Sensing*, vol. 11, no. 11, p. 1390, 2019.

[38] *Lossless Data Compression. Silver Book. Issue 1*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 121.0-B-1-S, May 1997. [Online]. Available: https://public.ccsds.org/Pubs/121x0b1sc2.pdf

[39] *Image Data Compression. Silver Book. Issue 1*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 122.0-B-1-S, May 2005. [Online]. Available: https://public.ccsds.org/Pubs/122x0b1c3s.pdf

[40] *Lossless Data Compression. Blue Book. Issue 2*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 121.0-B-2, Apr. 2012. [Online]. Available: https://public.ccsds.org/Pubs/121x0b2ec1s.pdf

[41] *Image Data Compression. Blue Book. Issue 2*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 122.0-B-2, Sep. 2017. [Online]. Available: https://public.ccsds.org/Pubs/122x0b2.pdf

[42] *Spectral Preprocessing Transform for Multispectral and Hyperspectral Image Compression. Blue Book. Issue 1*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 122.1-B-1, Sep. 2017. [Online]. Available: https://public.ccsds.org/Pubs/122x1b1.pdf

[43] *Lossless Data Compression. Blue Book. Issue 3*, Consultative Committee for Space Data Systems (CCSDS) Std. CCSDS 121.0-B-3, Aug. 2020. [Online]. Available: https://public.ccsds.org/Pubs/121x0b3.pdf

[44] D. Báscones, C. González, and D. Mozos, "Parallel implementation of the CCSDS 1.2.3 standard for hyperspectral lossless compression," *MDPI Remote Sensing*, vol. 9, no. 10, p. 973, 2017.

[45] A. Tsigkanos, N. Kranitis, G. A. 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*, pp. 1–1, 2018.

[46] 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.

[47] M. Orlandić, J. Fjeldtvedt, and T. A. Johansen, "A Parallel FPGA Implementation of the CCSDS-123 Compression Algorithm," *MDPI Remote Sensing*, vol. 11, no. 6, p. 673, 2019.

[48] L. M. V. Pereira, D. A. Santos, C. A. Zeferino, and D. R. Melo, "A low-cost hardware accelerator for ccsds 123 predictor in fpga," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[49] L. Santos, L. Berrojo, J. Moreno, J. F. López, and R. Sarmiento, "Multispectral and Hyperspectral Lossless Compressor for Space Applications (HyLoC): A Low-Complexity FPGA Implementation of the CCSDS 123 Standard," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 757–770, 2016.

[50] L. Santos, A. J. Gomez, and R. Sarmiento, "Implementation of CCSDS standards for lossless multispectral and hyperspectral satellite image compression," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1120–1138, 2020.

[51] Y. Barrios, A. J. 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.

[52] High-Speed Integrated Satellite Data Systems For Leading EU IndustrY, Hi-SIDE Project, H2020-COMPET-3-2017 (RIA): High speed data chain, European Commission, https://cordis.europa.eu/project/id/776151https://cordis.europa.eu/project/id/776151.

[53] Next Generation Satellite Processing Chain for Rapid Civil Alerts, EO-ALERT Project, H2020-COMPET-3-2017 (RIA): High speed data chain, European Commission, https://cordis.europa.eu/project/id/776311https://cordis.europa.eu/project/id/776311.

[54] D. Keymeulen, D. Dolman, S. Shin, J. Riddley, M. Klimesh, A. Kiely, D. R. Thompson, M. Cheng, S. Dolinar, E. Liggett, P. Sullivan, M. Bernas, K. Roth, C. Holyoake, K. Crocker, and A. Smith, "High Performance Space Data Acquisition, Clouds Screening and Data Compression with modified COTS Embedded System-on-Chip Instrument Avionics for Space-based Next Generation Imaging Spectrometers (NGIS)," in *6th International Workshop on On-Board Payload Data Compression (OBPDC)*, 2018.

[55] Copernicus Hyperspectral Imaging Mission for the Environment, "Mission Requirements Document," http://esamultimedia.esa.int/docs/EarthObservation/Copernicus_CHIME_MRD_v2.1_Issued20190723.pdfhttp://esamultimedia.esa.int/docs/EarthObservation/Copernicus_CHIME_MRD_v2.1_Issued20190723.pdf.

[56] M. Conoscenti, R. Coppola, and E. Magli, "Constant SNR, Rate Control, and Entropy Coding for Predictive Lossy Hyperspectral Image Compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7431–7441, 2016.

[57] J. Bartrina-Rapesta, I. Blanes, F. Aulí-Llinàs, J. Serra-Sagristà, V. Sanchez, and M. W. Marcellin, "A Lightweight Contextual Arithmetic Coder for On-Board Remote Sensing Data Compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4825–4835, 2017.

[58] J. Song, Z. Zhang, and X. Chen, "Lossless compression of hyperspectral imagery via RLS filter," *Electronics letters*, vol. 49, no. 16, pp. 992–994, 2013.

[59] F. Gao and S. Guo, "Lossless compression of hyperspectral images using conventional recursive least-squares predictor with adaptive prediction bands," *Journal of Applied Remote Sensing*, vol. 10, no. 1, p. 015010, 2016.

[60] A. C. Karaca and M. K. Güllü, "Lossless hyperspectral image compression using bimodal conventional recursive least-squares," *Remote Sensing Letters*, vol. 9, no. 1, pp. 31–40, 2018.

[61] ——, "Superpixel based recursive least-squares method for lossless compression of hyperspectral images," *Multidimensional Systems and Signal Processing*, vol. 30, no. 2, pp. 903–919, 2019.

[62] D. Keymeulen, S. Shin, J. Riddley, M. Klimesh, A. Kiely, E. Liggett, P. Sullivan, M. Bernas, H. Ghossemi, G. Flesch, M. Cheng, S. Dolinar, D. Dolman, K. Roth, C. Holyoake, K. Crocker, and A. Smith, "High Performance Space Computing with System-on-Chip Instrument Avionics for Space-based Next Generation Imaging Spectrometers (NGIS)," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Aug. 2018, pp. 33–36.

[63] M. Klimesh, "Low-Complexity Lossless Compression of Hyperspectral Imagery via Adaptive Filtering," Jet Propulsion Lab, NASA, Tech. Rep., 2005.

[64] D. Valsesia and E. Magli, "A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6341–6355, 2014.

[65] ——, "Fast and Lightweight Rate Control for Onboard Predictive Coding of Hyperspectral Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 394–398, 2017.

[66] R. Guerra, Y. Barrios, M. Díaz, A. Baez, S. López, and R. Sarmiento, "A Hardware-Friendly Hyperspectral Lossy Compressor for Next-Generation Space-Grade Field Programmable Gate Arrays," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 4813–4828, 2019.

[67] Consultative Committee for Space Data Systems (CCSDS), *Lossless Data Compression. Green Book. Issue 3*, 2013.

[68] I. Blanes, A. Kiely, L. Santos, M. Hernández-Cabronero, and J. Serra-Sagristà, "The Hybrid Entropy Encoder of CCSDS 123.0-B-2: Insights and Decoding Process," in *7th International Workshop on On-Board Payload Data Compression (OBPDC)*, Sep. 2020.

[69] M. Hernández-Cabronero, J. Portell, I. Blanes, and J. Serra-Sagristà, "High-Performance Lossless Compression of Hyperspectral Remote Sensing Scenes Based on Spectral Decorrelation," *MDPI Remote Sensing*, vol. 12, no. 18, p. 2955, 2020.

[70] E. Magli, G. Olmo, and E. Quacchio, "Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 1, pp. 21–25, 2004.

[71] F. A. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, "The spectral image processing system (SIPS)-interactive visualization and analysis of imaging spectrometer data," in *AIP Conference Proceedings*, vol. 283, no. 1. American Institute of Physics, 1993, pp. 192–201.