Received 13 February 2024, accepted 21 February 2024, date of publication 5 March 2024, date of current version 13 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3374227

TUTORIAL

Resiliency and Efficiency of the CCSDS 124.0-B-1 Telemetry Compression Standard

MIGUEL HERNÁNDEZ-CABRONERO[1], DAVID EVANS[2], JOAN BARTRINA-RAPESTA[1], FRANCESC AULÍ-LLINÀS[1], (Senior Member, IEEE), IAN BLANES[1], AND JOAN SERRA-SAGRISTÀ[1], (Senior Member, IEEE)

[1]Information and Communication Engineering (DEIC), Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain
[2]ESA/ESOC, 64293 Darmstadt, Germany

Corresponding author: Miguel Hernández-Cabronero (miguel.hernandez@uab.cat)

This work was supported in part by Beatriu de Pinós Programme funded by the Secretary of Universities and Research (Government of Catalonia) under Project 2018-BP-00008 and Project 2021 SGR 00643; in part by the Horizon 2020 under the Marie Skłodowska-Curie under Grant 801370; in part by Spanish Ministry of Science and Innovation and the European Regional Development Fund under Grant RTI2018-095287-B-I00; in part by French Space Agency Centre National d'Etudes Spatiales (CNES) under Grant DSO/TB/ET/2019-13107 and Grant DSO/TB/ET/2020-29135; and in part by MCIN/AEI/10.13039/501100011033/FEDER, European Union (UE), under Project PID2021-125258OB-I00.

ABSTRACT The Consultative Committee for Space Data Systems (CCSDS) recently published a lossless compression standard for housekeeping and telemetry. These data are critical for the safe and productive operation of virtually all remote sensing missions, including Earth observation. The new standard, CCSDS 124.0-B-1 "Robust Compression of Fixed-Length Housekeeping Data," allows lossless, real-time telemetry packet transmission with low energy and memory requirements. Furthermore, it offers flexible resiliency mechanisms to resume decompression after losing one or more data packets. This work provides an accessible introduction to CCSDS 124.0-B-1, paying special attention to its resiliency features. Two novel and complementary decoder resynchronization strategies are also presented, which maximize the amount of correctly decoded data. Comprehensive experiments with real and simulated mission data were conducted for this work to assess the standard's performance and the impact of user-defined parameters. Results indicate competitive coding performance, with average compression ratios between 4 and 16 depending on the dataset. Smooth trade-offs are also observed between compression performance and the degree of protection against data loss events, which enhances the usability of CCSDS 124.0-B-1 in multiple mission types and operational conditions.

INDEX TERMS Data compression, telemetry, space communications.

I. INTRODUCTION

Housekeeping telemetry is defined as the data fraction produced on board space and remote sensing systems for the purpose of monitoring their correct functioning. These data are usually composed of values periodically registered by different types of sensors, including temperature, pressure, attitude, and orbit control measurements [1]. Reliable and timely reception of these data is paramount to guarantee nominal system behavior, as well as its safety and that of surrounding spacecraft [2].

Space transmission links offer limited bandwidths and are often available only intermittently. Furthermore, ground station use is traditionally charged by time, and the energy required to maintain a link can be severely limited in small satellites such as nanosats and cubesats [3]. Currently, the produced amount of housekeeping data is experiencing a steep increment due to the larger number of monitored variables and the use of higher sampling rates [4]. To compensate for these limitations, data compression enables more efficient utilization of the available transmission channels, lower latency and energy usage, allowing a larger channel bandwidth fraction to be used for payload data, as well as lowering operational costs. These advantages can be

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Di Martino.

particularly significant for missions with long cruise periods, technology demonstrators, and small satellites [3].

Two key aspects determine the practicality and benefits of compressing housekeeping data: coding efficiency and resiliency to data loss. Algorithms that produce smaller coded data volumes are more advantageous in terms of bandwidth usage, transmission latency, and operational costs. Notwithstanding, compression must comply with the computational and energetic constraints imposed by the mission [2], [3], [5]. Lossless compression is also considered a requirement in many remote sensing mission types [6], [7], [8]. In addition, the system must continue to work even when part of the compressed data stream is lost, because automatic repeat/retransmission queries are not always possible, supported by the spacecraft systems, or feasible in terms of energy and computational resources [9]. Therefore, compression algorithms with high coding efficiency, but which introduce dependencies between successive parts of the coded data [4], [5], [7], [8], can make them unsuitable in these scenarios because these dependencies can prevent the decoder from reliably reconstructing the original data from the data stream after a single data loss event. This issue needs to be carefully addressed.

It is worth noting that data loss events may happen even in the presence of strong forward error correction mechanisms at the data link layer, *e.g.*, those described in [11], [12], [13], [14], and [15]. Although most space links are designed to offer frame error rates between 10^{-7} and 10^{-5}, data losses do occur in practice due to low elevations, very bad weather, operator errors and intermittent contact during contingency situations such as safe mode [1], [13]. Thus, it is necessary to employ algorithms that allow the decompression of incomplete compressed data streams.

The Consultative Committee for Space Data Systems (CCSDS) is a consortium of the most important space agencies on the planet. Its main goal is the creation of free, publicly available standards to support space mission interoperability. Recently, the CCSDS has completed the definition of the CCSDS 124.0-B-1 standard [16], titled "Robust Compression of Fixed-Length Housekeeping Data." It aims at providing efficient, real-time, lossless compression of housekeeping telemetry, while incorporating configurable types and amounts of resiliency to data loss and corruption.

The main contributions herein presented are a comprehensive yet accessible description of the CCSDS 124.0-B-1 standard, two novel decoder resynchronization strategies, and a thorough performance analysis, including optimal parameter selection and trade-offs between coding efficiency and resiliency. This paper does not presume any prior knowledge of this or any other CCSDS standard, and requires only a minimal understanding of the notions of data compression.

The remainder of this document is structured as follows. Section II reviews the most relevant works related to this standard. Section III describes the key aspects of CCSDS 124.0-B-1, including the proposed resynchronization strategies. Experimental results and an analysis of the aforementioned trade-offs are presented and discussed in Section IV. Finally, Section V concludes this paper with the most important remarks.

II. PREVIOUS WORK

Spacecraft housekeeping data is usually transmitted in packets where only the value of the monitored variable values are written. The ground allocates these values to the corresponding variables by knowing their positions in the packet, *i.e.*, each packet has a fixed length and structure. Also, the monitored values tend to change slowly and sometimes periodically over time. State-of-the-art works exploit these characteristics to maximize their coding efficiency. Some methods divide the input packets into groups, which are independently coded. This provides some resilience as the decompression process can continue for other groups after a data loss event thanks to the lack of coding dependencies between the groups. In [6], bitwise exclusive-or (XOR) operations are applied periodically within the group, and then a modified version of run-length encoding (RLE) is proposed to encode the resulting data. In [1] and [3], it is shown that lossless compression methods such as Lempel-Ziv (LZ) and RLE yield significantly better compression results when applied to individual bit positions across several data packets. In [4], groups are dynamically created based on a similarity metric, and then the differences between consecutive packets in a group are coded with either RLE or LZ. An extension to [4] using parallel string search and a modified LZ algorithm is proposed in [17]. Recently, in [18], dynamic bit packing is applied to time series after applying XOR operations to successive data samples from the same sensor.

Despite these benefits, the main drawback of packet grouping is that data packets cannot be sent in real time, *i.e.*, as soon as they are produced, because they need to be jointly processed to yield competitive compression performance. This shortcoming may not be tolerable in scenarios where prompt reception of system status data is critical, or when telemetry packets are transmitted with low frequency.

Several methods have been proposed that allow real-time packet transmission. A lossless compression algorithm is proposed in [5] that performs bytewise comparisons between each packet and the previous one. The encoded data contain only position-value pairs for those bytes that do change, but no resiliency mechanism is considered for data loss events. In [19], a low-complexity method for compressing individual signals is proposed that computes the differences between consecutive samples, and then uses 1, 2, 3 or 4 byte codewords depending on their absolute value. This method, although fast, offers lower compression performance than other algorithms, and also does not provide data loss resiliency mechanisms. In [8], an adaptive version of differential pulse-code modulation (DPCM) is applied to consecutive packets, and the result is stored using a variable number of bits. Resiliency is provided by periodically

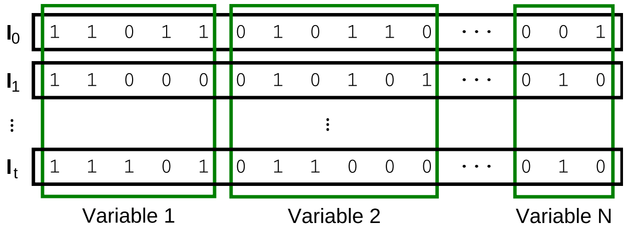

FIGURE 1. Structure of a sequence of t uncompressed packets. Illustrative data are provided for the contents of three packets.

sending reference data not processed with DPCM. The CCSDS 124.0-B-1 standard, described in detail in the next section, is directly based on the POCKET and POCKET+ algorithms proposed in [7] and [10]. A summary of other housekeeping telemetry compression algorithms is provided in Table 1 for ease of reference.

Other housekeeping telemetry compression algorithms have been proposed that entail high computational costs [20], [21], [22], [23], or that provide only lossy compression [24], [25], [26]. Therefore, these methods lie outside the scope of this work.

III. THE CCSDS 124.0-B-1 STANDARD

The CCSDS 124.0-B-1 standard offers lossless, real-time compression of sequences of housekeeping telemetry packets of identical length. Lossless compression guarantees perfect reconstruction of the transmitted packets, which is of utmost importance for safety-critical telemetry data. Lossy compression is not supported by the standard, but users may apply any desired pre-processing to the monitored variables before inputting them to the compressor, e.g., quantization of sensor readings. The real-time capabilities of the standard allow compressing and transmitting each packet as soon as it is created. This prevents unnecessary delays for safety-related telemetry, and also for sensor data acquired with low periodicity.

Packets may include any number of monitored variables, *e.g.*, temperature, pressure, attitude, etc. The number of variables, their precision and their binary representation inside the packets are arbitrarily defined by the user to accommodate for all types of missions and telemetry data. No specific information about the included monitored variables or their representation format needs to be transmitted or known by the encoder or decoder implementations. A diagram that illustrates the structure of a sequence of uncompressed packets is shown in Fig. 1.

Section III-A explains the normal operation of the compression algorithm and the principle upon which its efficiency is based. Section III-B describes the resiliency mechanisms provided by the standard to allow decompression after one or more packets are lost or corrupted during transmission. Section III-C proposes a novel algorithm that minimizes the resynchronization latency after a data loss event and maximizes the amount of correctly decodable data.

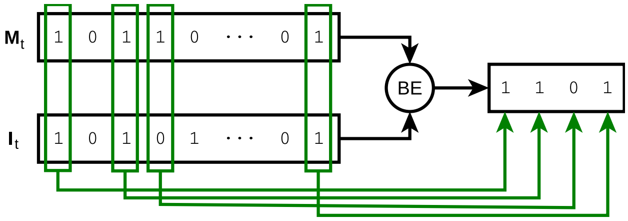

FIGURE 2. Bit extraction (BE) of unpredictable positions.

In what follows, bit vectors are represented in bold font —*e.g.*, $\mathbf{I}_t$—, scalar integer and boolean variables in italics —*e.g.*, R_t—, and individual bit values with `0` and `1`. The vectors containing only zeros and only ones are denoted with $\mathbf{0}$ and $\mathbf{1}$, respectively. When applicable, the discrete time index t of a vector or a scalar value is represented as a subscript. In the interest of readability, a table of symbols defined in CCSDS 124.0-B-1 and in the standard is provided in the Appendix.

A. NORMAL OPERATION

The compression performance of CCSDS 124.0-B-1 is based on the fact that most telemetry variables change slowly over time. Under normal operation, the compressor considers the bitwise differences between consecutive packets —*i.e.*, $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$, interpreted as bit vectors— instead of the full packet contents. When the slow variable change assumption is satisfied, these differences are highly compressible. Coding efficiency is maximized when all packets in a sequence contain the same monitored variables in the same order and using the same data format, as previously shown in Fig. 1.

To exploit the similarities among consecutive packets, the standard keeps track of an auxiliary variable $\mathbf{M}_t$ called the *mask* vector. When the packet $\mathbf{I}_t$ at index t is received, the mask is updated as

$$\mathbf{M}_t = \mathbf{M}_{t-1} \text{ OR } (\mathbf{I}_{t-1} \text{ XOR } \mathbf{I}_t)\,. \tag{1}$$

Bit positions set to `1` in the mask are considered *unpredictable*, while positions set to `0` are known not to have changed in $\mathbf{I}_t$ with respect to previous packets. Based on this, the compressor applies a *bit extraction* (BE) operation to the uncompressed packet and transmits packet bits at unpredictable positions, as illustrated in Fig. 2. In addition, accumulated changes in the mask are transmitted using run-length encoding (RLE), as detailed in Section III-B. These two pieces of information are sufficient for the decoder to losslessly reconstruct the original packet in the absence of packet losses. Under normal conditions, their combined length is significantly shorter than that of the original packets, resulting in competitive compression ratios. It is worth stressing that this design allows both efficient and real-time transmission of the packets, *i.e.*, packet buffering is not needed in order to exploit the similarities of consecutive packets.

When Eq. (1) is applied, mask bit positions can only transition from predictable to unpredictable. After processing

TABLE 1. Summary of related works. CCSDS 124.0-B-1 is highlighted in bold font.

Reference	Year	Coding approach	Robustness approach	Real time transmission
[6]	2000	XOR, Run-Length Encoding	Grouping	No
[1]	2010	Lempel-Ziv	Grouping	No
[4]	2018	Run-Length Encoding, Lempel-Ziv	Grouping	No
[17]	2022	Run-Length Encoding, Modified Lempel-Ziv	Grouping	No
[18]	2023	XOR, Dynamic bit packing	Grouping	No
[5]	2013	Difference calculation, Position-value signaling	None	Yes
[8]	2016	Differential Pulse-Code Modulation	Reference data	Yes
[19]	2021	Difference calculation, Variable-length byte words	None	Yes
[16]	**2023**	**XOR** **Predictability mask** **Run-Length Encoding**	**Robustness level** **Reference data** **Partial synchronization**	**Yes**

enough packets in this fashion, most positions in the mask are flagged as unpredictable, which can severely diminish the observed coding efficiency. To avoid this problem, the standard defines a *build mask* vector $\mathbf{B}_t$ and a *new mask* flag p_t. When the user activates this flag, the mask and build mask vectors are updated as follows:

$$\begin{aligned} \mathbf{M}_t &= \mathbf{B}_{t-1} \text{ OR } (\mathbf{I}_{t-1} \text{ XOR } \mathbf{I}_t) \\ \mathbf{B}_t &= \mathbf{0}. \end{aligned} \tag{2}$$

Otherwise, the build mask is updated as in (1), *i.e.*,

$$\mathbf{B}_t = \mathbf{B}_{t-1} \text{ OR } (\mathbf{I}_{t-1} \text{ XOR } \mathbf{I}_t)\,. \tag{3}$$

As a result, the mask vector only flags those positions as unpredictable if they changed state in the input vectors between successive activations of the new mask flag p_t. Activating it periodically allows the encoder to exploit slowly changing nature of telemetry data. The effect of this flag on the standard's compression efficiency is analyzed in Section IV-B.

B. RESILIENCY MECHANISMS

Resuming decompression of a compressed data stream after data loss events is critical for the reliable acquisition of telemetry data in remote sensing. In particular, when retransmission is not supported or practical for a mission, decompression must be possible even with incomplete or partially corrupted data. The CCSDS 124.0-B-1 standard defines two mechanisms to enhance its resiliency to packet losses, imposing two requirements on the communication link transport layer. First, that each transmitted packet is accompanied by its correct index t within the sequence of transmitted packets. Second, that corrupted packets whose errors cannot be corrected are flagged as lost.

1) ROBUSTNESS LEVEL

The first resiliency mechanism is not present in any other algorithm in the state of the art. For each packet $\mathbf{I}_t$, the user selects its *minimum required robustness level*, $0 \leq R_t \leq 7$. For $R_t > 0$, the standard guarantees that as long as at least one of the previous R_t packets $(\mathbf{I}_{t-R_t}, \mathbf{I}_{t-R_t+1}, \ldots, \mathbf{I}_{t-1})$ was correctly decompressed, I_t can be losslessly decompressed upon its correct reception. To do so, the *change vector* $\mathbf{D}_t$ and *accumulated change* vector $\mathbf{X}_t$ are defined as

$$\begin{aligned} \mathbf{D}_t &= \mathbf{M}_t \text{ XOR } \mathbf{M}_{t-1} \\ \mathbf{X}_t &= \mathbf{D}_t \text{ OR } \mathbf{D}_{t-1} \text{ OR } \cdots \text{ OR } \mathbf{D}_{t-R_t}, \end{aligned} \tag{4}$$

i.e., $\mathbf{X}_t$ contains `1` only at those positions where the mask has changed in any of the last R_t packets. Vector $\mathbf{X}_t$ is transmitted, along with all bits of the mask $\mathbf{M}_t$ where $\mathbf{X}_t$ is `1`. Furthermore, the bit extraction of $\mathbf{I}_t$, described in Section III-A, is performed at all positions where either $\mathbf{M}_t$ or $\mathbf{X}_t$ are `1`. This additional information allows the decoder to reconstruct $\mathbf{I}_t$, $\mathbf{M}_t$, and to decode any subsequent, correctly received packet. The impact of the minimum required robustness level R_t on the standard's compression performance is analyzed in Section IV-C.

An important observation about $\mathbf{X}_t$ is the fact that $\mathbf{D}_t = \mathbf{0}$ implies $\mathbf{X}_t = \mathbf{X}_{t-1}$. In this scenario, if the user selects a minimum required effective robustness level $R_t = 0$, the compressed packet contains the same information as if $R_t = 1$ had been selected, without any compression performance penalty. More generally, if $\mathbf{D}_{t-R_t-1} = \mathbf{D}_{t-R_t-2} = \cdots = \mathbf{D}_{t-R_t-C_t} = \mathbf{0}$ for some $C_t \geq 0$, then the decoder can reconstruct $\mathbf{I}_t$ whenever one of the previous

$$V_t = R_t + C_t \tag{5}$$

packets could be correctly deconstructed, instead of one of the previous R_t. Based on this, the CCSDS 124.0-B-1 compressor signals V_t in each compressed packet, known as the *effective robustness level*, instead of R_t. In other words, the standard analyzes the recent history of changes to the mask to guarantee the maximum robustness to data loss events, without affecting the observed compression ratio. The relation between the minimum requested effective robustness level, R_t, and the actual robustness observed by the decoder, V_t, is also analyzed in Section IV-C.

2) FULL MASK AND UNCOMPRESSED PACKET FLAGS

The second resiliency mechanism allows the user to encode the contents of the mask $\mathbf{M}_t$ and of the uncompressed packet $\mathbf{I}_t$ instead of only the mask changes and the packet bits at unpredictable positions. This is done, respectively, by activating the *full mask* flag f_t and the *uncompressed* flag r_t.

When both are simultaneously activated, hereafter referred to as transmitting a *full reference* packet, any dependencies with previous packets are broken. Thus, the compressed packet becomes self-contained and can be decoded upon correct reception regardless of whether previous packets could be reconstructed. This is analogous to [8], which interleaves the transmission of packet differences and full packet contents. In terms of resiliency to data loss events —but not real-time transmission capabilities—, periodically transmitting full reference packets is roughly equivalent to [1], [3], [4], [6], [17], and [18] when packets are divided in groups and compressed separately. The effect on compression efficiency of periodically activating both the full mask and uncompressed flags is analyzed in Section IV-D.

C. PROPOSED RESYNCHRONIZATION STRATEGIES

When one or more packets are lost or flagged as corrupted by the transport layer, the decoder loses *synchronization* with the encoder. Packets successfully transmitted after the data loss events can only be correctly decompressed after resynchronization takes place. As described in Section III-B, this can occur automatically thanks to the effective robustness level V_t and the use of full reference packets where f_t and r_t are simultaneously activated. Notwithstanding, thanks to the flexibility of user-defined parameters and the compressed packet structure in CCSDS 124.0-B-1, these are not the only ways to attain resynchronization. Two novel, complementary resynchronization strategies are hereafter proposed, that can be implemented within a standard-compliant decoder.

1) PARTIAL RESYNCHRONIZATION

Instead of simply using an all-or-nothing approach to synchronization, the decoder defines two status flags for each index in the packet sequence: the *synchronized mask* flag s_t^M and the *synchronized packet* flag s_t^I. Flag s_t^M is enabled if and only if the decoder knows the value of all bit positions of the mask vector M_t. Conversely, flag s_t^I is enabled if and only if the decoder could guarantee lossless reconstruction of the input packet I_t. Synchronization with the encoder is then defined as the state in which both s_t^M and s_t^I are enabled, while *partial synchronization* occurs when only one of these two flags is enabled.

The proposed strategy attempts to restore synchronization of the mask and the reconstructed packet separately, and as soon as enough information is available. This allows the decoder to reach full resynchronization sooner and more often than the naive decompression algorithm. In particular, resynchronization becomes possible even if the needed required information is distributed across more than one compressed packet, *e.g.*, when the full mask flag f_t and uncompressed flag r_t are activated for different packets. An analysis of the impact of using different periodicities for f_t and r_t is included in Section IV-D.

To implement the partial synchronization strategy, the decoder defines two additional integer counters, the unsynchronized mask counter L_t^M and the unsynchronized packet counter L_t^I. When s_t^M is enabled, $L_t^M = 0$. Similarly, when s_t^I is enabled, $L_t^I = 0$. Otherwise, they are equal to, respectively, the number of consecutive, previous packets for which s_t^M and s_t^I where disabled. For instance, $L_t^M = 2$ indicates that $\mathbf{M}_{t-3}$ was known to the decoder (s_{t-3}^M was enabled), but not $\mathbf{M}_{t-2}$, $\mathbf{M}_{t-1}$ nor $\mathbf{M}_t$ (s_{t-2}^M, s_{t-1}^M, s_t^M were disabled).

Using this notation, the proposed approach can be completely described with the set of rules that enable s_t^M, and the set of rules that enable s_t^I. Mask synchronization (s_t^M enabled and $L_t^M = 0$) takes place whenever the t-th compressed packet is correctly received and any of the following conditions is met:

- A-1) $V_t \geq L_t^M$.
- A-2) f_t is enabled.
- A-3) s_{t-1}^M is enabled.
- A-4) $V_t > 0$ and $\mathbf{X}_t = \mathbf{1}$.

In turn, packet synchronization (s_t^I enabled and $L_t^I = 0$) occurs whenever the t-th compressed packet is received and any of the following is true:

- B-1) $V_t \geq L_t^I$.
- B-2) r_t is enabled.
- B-3) Both s_{t-1}^M and s_{t-1}^I are enabled.

Conditions A-1) and B-1) can be applied thanks to the robustness level mechanism described in Section III-B1. Conditions A-2) and B-2) are applicable due to the transmission of reference information, as described in Section III-B2. Condition A-3) means that once the mask is synchronized, it remains synchronized until the next data loss event. However, condition B-3) is asymmetrical and requires both the previous mask and reconstructed vector to be known. This is because the bits extraction operation described in Section III-A can only be reverted if the values of $\mathbf{M}_t$ are known. Finally, condition A-4) applies because, although rare, it results in a compressed packet containing the full contents of $\mathbf{M}_t$.

2) BACKWARDS DECODING

The resiliency and resynchronization mechanisms described in Sections III-B and III-C1 determine whether a compressed packet can be correctly decoded the moment it is successfully received. Under certain circumstances, packets that are undecodable at the time of reception become decodable after the correct reception of subsequent packets. A strategy called *backwards decoding* is herein proposed that exploits this fact and maximizes the number of packets in a sequence that can be correctly reconstructed.

After receiving the t-th packet whose corresponding mask $\mathbf{M}_t$ is synchronized —*i.e.*, s_t^M is enabled—, meeting any of the following conditions makes it possible to deduce $\mathbf{M}_{t-1}$ from it:

- If $V_t = 0$, then Eqs. (4) and (5) imply that

$$\mathbf{M}_{t-1} = \mathbf{M}_t \text{ XOR } \mathbf{X}_t. \tag{6}$$

- Similarly, if $\mathbf{X}_t = \mathbf{0}$, then (4) implies that

$$\mathbf{M}_{t-1} = \mathbf{M}_t. \tag{7}$$

The proposed strategy repeats this inference for $t-1$, $t-2, \ldots, t'$, until none of these conditions are satisfied for $t'-1$, packet at $t'-1$ was lost or corrupted, all packets between $t'-1$ and the previous data loss event were correctly decoded, or backwards decoding was already applied for the packet at time $t'-1$. Once the mask is propagated backwards in this fashion, synchronized mask flags $s^M_{t-1}, s^M_{t-2}, \ldots, s^N_{t'}$ are all enabled. After that, decompression of packets $t', t'+1, \ldots, t-1$ is attempted sequentially using all mechanisms described in Sections III-A, III-B and III-C1, updating s^I, L^M and L^I as necessary.

IV. EXPERIMENTAL RESULTS

The following experiments analyze the coding efficiency of CCSDS 124.0-B-1, as well as the impact of all user-defined parameters, when this algorithm is applied to different telemetry corpora. The compression ratio for each sequence of housekeeping telemetry packets is used as the figure of merit for coding efficiency. It is defined as

$$\text{Compression ratio} = \frac{\text{Original sequence size}}{\text{Compressed sequence size}}, \tag{8}$$

with larger values indicating better compression. All results are produced using a prototype developed for the French space agency (CNES), which has been cross-validated for compliance against another implementation by the European Space Agency (ESA). A list of publicly available reference implementations of the standard can be found in [27].

A. TEST CORPORA

Experiments are conducted on four corpora of housekeeping telemetry data. These were defined by the CCSDS Data Compression working group as a part of the standardization process of CCSDS 124.0-B-1.

- The *IceCube* corpus contains samples generated by a NASA cubesat using a commercial cloud radiometer for ice detection [28].
- The *Jason2* corpus consists of data produced by a NASA/CNES earth-observation satellite used for sea surface height measurements [29].
- The *PUS* corpus comprises synthetic packets simulated by the SimuPUS software developed by CNES to experiment new technologies.
- The *Venus* corpus contains samples from the Venus Express mission of ESA [30].

The corpus name, number of packet sequences, total size in bytes, average entropy in bits per sample and the range of housekeeping packet lengths in bits are summarized in Table 2. The dataset can be obtained in [31].

B. IMPACT OF THE NEW MASK FLAG P_T

The impact on coding efficiency of the new mask flag p_t is analyzed first to find optimal values for this user-defined parameter. Recall that this parameter does not affect the resiliency to packet loss events.

TABLE 2. Properties of the test corpora.

Corpus	Count	Total size	Entropy	Packet lengths
IceCube	1	8 089 676	2.867	1 376 – 1 376
Jason2	11	5 738 213	5.376	232 – 3 416
PUS	12	8 225 250	1.941	1 384 – 8 192
Venus	11	65 691 090	5.336	432 – 4 016

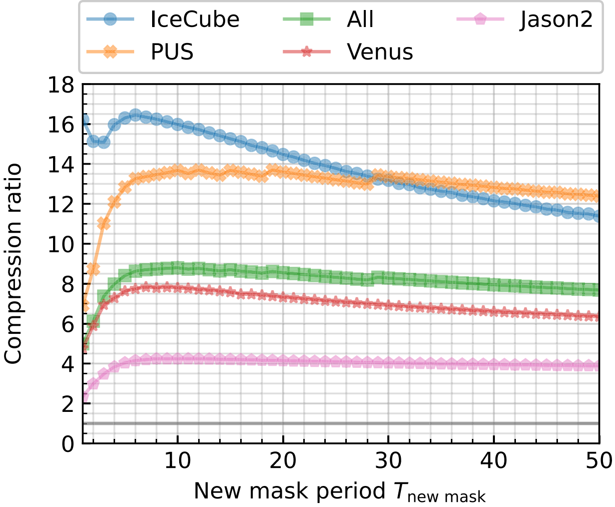

FIGURE 3. Average compression ratios for each corpus and for all test sequences as a function of the new mask period.

The CCSDS 124.0-B-1 encoder is applied activating the new mask flag p_t once every $T_{\text{new mask}}$ packets, where $T_{\text{new mask}} \in \{1, 2, \ldots, 50\}$ is the *new mask period*. The minimum required effective robustness level R_t is set to 0 for all packets. The full mask flag f_t and the uncompressed flag r_t are not activated except for the first packet of each sequence, as mandated by the standard. Sequences are independently compressed and their compression ratio is calculated via (8). The average compression ratios for each corpus and each new mask period are shown in Fig. 3.

In all cases, the highest coding performance is obtained for relative small new mask periods ranging between 5 and 10, depending on the corpus. Smaller values of $T_{\text{new mask}}$, *i.e.*, more frequent activations of the p_t flag, cause steep performance penalties. This is explained by the fact that the build vector is reset very frequently, and the mask vector is replaced by it only a few packets later according to (2). As a result, relatively few bit positions of the mask vector are set to `1`, limiting the ability of the mask vector to predict what positions change in consecutive packets. Therefore, the change vector $\boldsymbol{D}_t$ defined in (4) contains a relatively large number of positions set to `1`, which increases its encoding cost. The main exception to this trend is found with the IceCube corpus, in which activation of the new mask flag p_t in every packet yields compression ratios similar to those of the best value of $T_{\text{new mask}}$. This is explained by the higher prevalence of bits set to `0` in this corpus, which reduces the

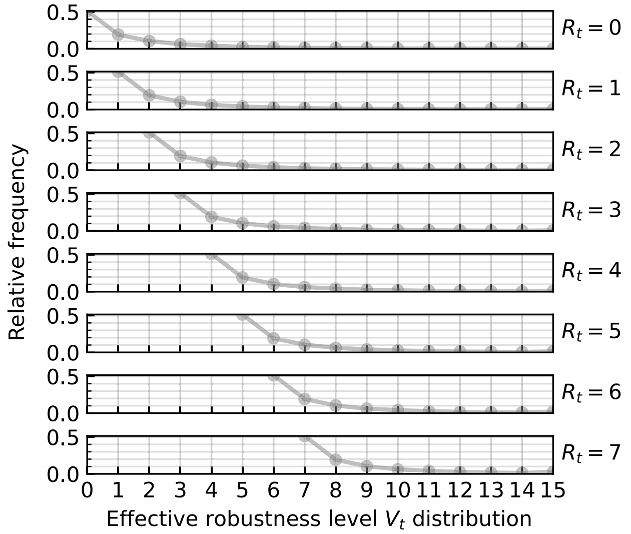

FIGURE 4. Distribution of average effective robustness level V_t for all test sequences as a function of the minimum required effective robustness level.

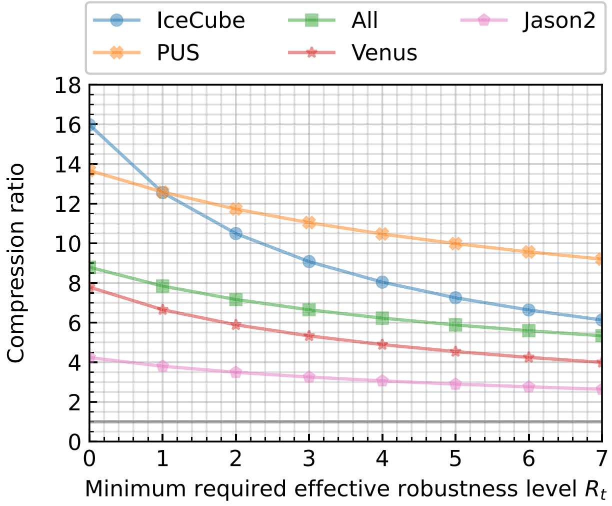

FIGURE 5. Average compression ratios for each corpus and for all sequences as a function of the minimum required effective robustness level.

cost of coding $\boldsymbol{D_t}$. On the other hand, increasing $T_{\text{new mask}}$ beyond the optimum causes a slow coding performance deterioration. This is due to the build vector being reset to $\mathbf{0}$ and used to replace the mask vector only occasionally. In this case, the mask vector tends to flag most bit positions as unpredictable, which in turn increases the cost of encoding the bits at these positions.

C. IMPACT OF THE MINIMUM REQUIRED EFFECTIVE ROBUSTNESS LEVEL R_T

The following tests address the impact of the minimum required effective robustness level R_t in both resiliency against packet loss events and coding efficiency.

As discussed in Section III, decompression of a received packet is guaranteed to be successful as long as any of the previous $V_t \geq R_t$ packets was successfully decoded, where V_t is the effective robustness level. The value of V_t is not directly chosen by the user, and must be evaluated experimentally. To do so, CCSDS 124.0-B-1 is configured to activate the new mask flag p_t periodically every $T_{\text{new mask}} = 10$ packets, while the full mask flag f_t and uncompressed flag r_t are only activated for the first packet of each sequence. The average distribution of V_t for all packet sequences is computed for all possible values of R_t and depicted in Fig. 4. In all cases, between 51.4% and 51.5% of the packets satisfy $V_t = R_t$, while additional resiliency is added to the remaining packets. More specifically, V_t exceeds R_t by 1, 2 or 3 respectively in about 19%, 11% and 6% of the packets. Larger resiliency increments are less frequent, but the maximum effective robustness value $V_t = 15$ is attained in 0.2%–0.3% of the packets for all values of R_t.

The effect of R_t on coding efficiency is also analyzed. To do so, the same configuration of CCSDS 124.0-B-1 is employed. Fig. 5 shows the average compression ratios obtained for each corpus and each possible value of R_t. Larger values of R_t produce lower compression ratios, due to the additional information included by the encoder. Compared to $R_t = 0$, average reductions between 8% and 21% are observed for $R_t = 1$, and between 33% and 62% for $R_t = 7$. It can also be observed that increasing R_t by 1 yields smaller penalties for larger values of R_t. This is due to the slowly changing nature of housekeeping telemetry data, which results in comparatively fewer bits changed to 1 in $\boldsymbol{X_t}$ as per (4). Also note that given a value of R_t, coding efficiency is not affected by V_t.

D. IMPACT OF THE FULL MASK FLAG F_T AND UNCOMPRESSED FLAG R_T

The effect of the full mask flag f_t and the uncompressed flag r_t is evaluated next when they are activated with identical and different periodicities.

When these two flags are simultaneously activated and the packet is successfully received, the decoder can decompress the packet even if the synchronization with the encoder was broken due to a previous data loss event. Furthermore, this synchronization is restored, *i.e.*, successive packets may be decoded until the next data loss, event if f_t and r_t are not activated in successive packets. The simultaneous activation of f_t and r_t comes at the cost of reduced coding efficiency, due to the additional information sent to the decoder. To quantify this penalty, CCSDS 124.0-B-1 is applied by activating both f_t and r_t once every $T_{\text{full ref}}$ packets, where $T_{\text{full ref}} \in \{1, 2, \ldots, 30\}$ is the *full reference period*. To maximize coding efficiency, the new mask flag p_t is periodically activated with period $T_{\text{new mask}} = 10$, and the minimum required effective robustness level R_t is set to 0 for all packets. Results for this experiment are shown in Fig. 6. When f_t and r_t are activated every packet, *i.e.*, $T_{\text{full ref}} = 1$, compression ratios drop below 1 for all corpora, that is, data expansion is produced. For all other values of $T_{\text{full ref}}$, average compression ratios above 1 are observed. As the

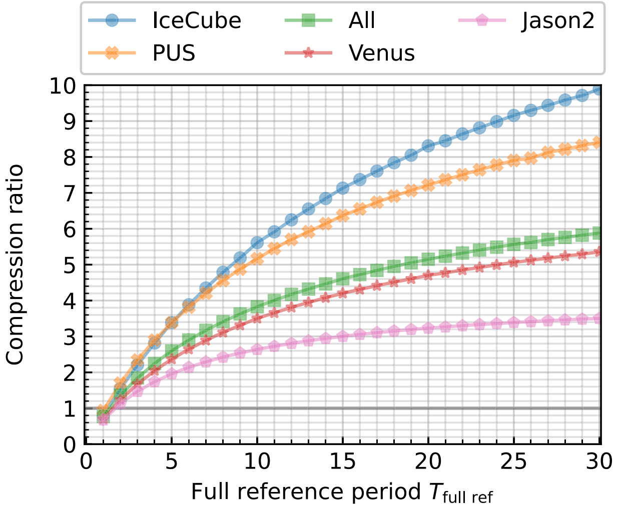

FIGURE 6. **Average compression ratios for each corpus and for all sequences as a function of the full-reference period.**

value of $T_{\text{full ref}}$ is increased, the coding efficiency penalties are reduced proportionally to the inverse of $T_{\text{full ref}}$. Note that in some cases, $T_{\text{full ref}} = T$ yields better results than $T_{\text{full ref}} = T + 1$, due to the cost variability of encoding the result of sending the full mask for different packets.

With the proposed strategies discussed in Section III-C, resynchronization can take place even when the full mask flag f_t and uncompressed flag r_t are not enabled for the same packets. In the next experiment, the impact on the compression ratio is evaluated when f_t and r_t are activated with periods $T_{\text{full mask}}$ and $T_{\text{uncompressed}}$, respectively. Other parameters are unchanged, *i.e.*, $R_t = 0$ and $T_{\text{new mask}} = 10$. In Fig. 7, results are shown for $T_{\text{full mask}} = kT$, $T_{\text{uncompressed}} = T$, where $k \in \{2, 3, 4\}$ and $T \in \{1, 2, \ldots, 30\}$, *i.e.*, the full mask flag is activated k times less often (period k times larger) than the uncompressed mask flag. In Fig. 7b, results are shown for $T_{\text{full mask}} = T$, $T_{\text{uncompressed}} = kT$ for the same values of k and T, *i.e.*, the full mask flag is activated k times more often than the uncompressed flag. In both cases, results are expressed as the difference with $T_{\text{full mask}} = T$, $T_{\text{uncompressed}}$, *i.e.*, $T_{\text{full ref}} = T$. Data suggest that reducing the activation frequency of either flag results in non-negligible compression ratio increments. Observed increments are particularly striking when $T_{\text{uncompressed}}$ is activated less frequently, which stem from the fact that transmitting an uncompressed packet is more expensive than signaling the full mask. This can be explained by the fact that, due to the slowly changing rate of remote sensing telemetry variables, the mask contains a low fraction of bits set to 1 and the applied run-length encoding is highly efficient.

E. COMPARISON TO OTHER METHODS

The coding efficiency and compression time of CCSDS 124.0-B-1 is compared to a selection of lossless compression algorithms. CCSDS 124.0-B-1 is configured to use $R_t = 0$, to activate the new mask flag p_t every 10 packets, and to activate the full mask and uncompressed flags only in the first packet of each sequence. The "Lossless Data Compression" standard [32], based on Golomb-Rice coding and RLE,

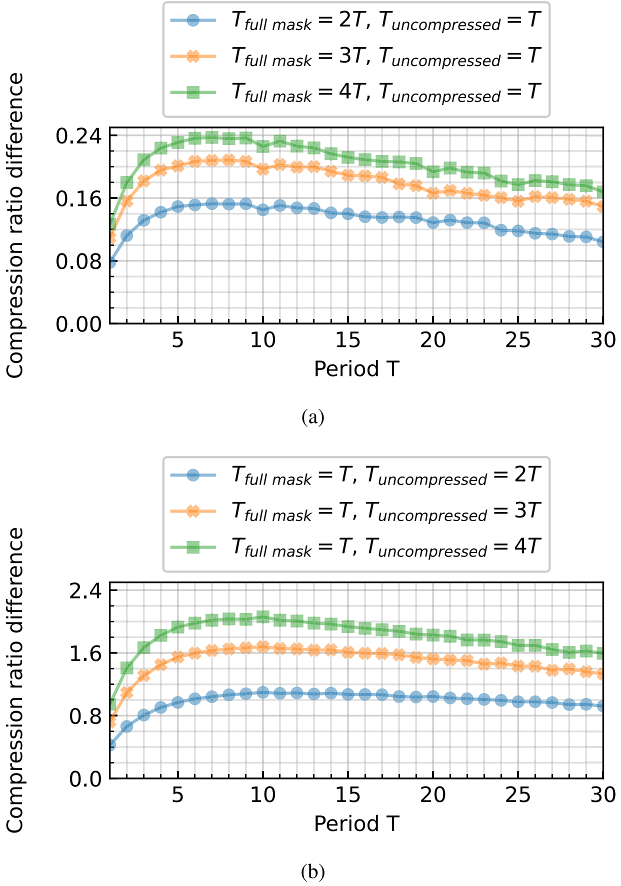

FIGURE 7. **Average compression ratio differences for all sequences with respect to the full-reference approach shown in Fig. 6, when (a) transmitting full masks less frequently; and (b) when transmitting uncompressed packets less frequently.**

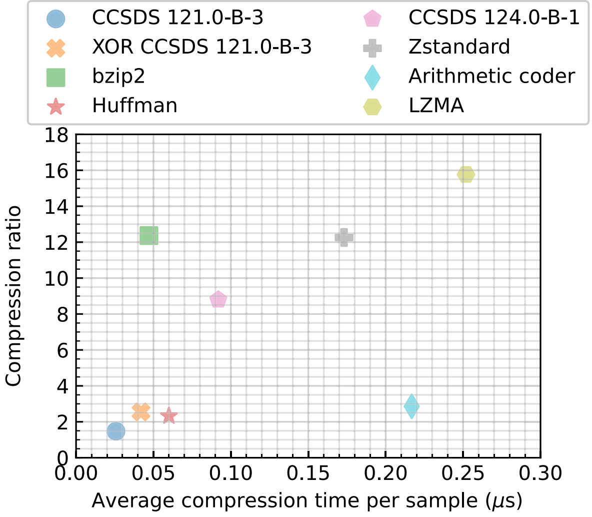

FIGURE 8. **Average compression ratios and encoding times per sample for all test sequences.**

is applied directly (labeled *CCSDS 121.0-B-3*) and after the XOR of consecutive packets (labeled *XOR CCSDS 121.0-B-3*). In both cases, a block size of 64 samples and a reference period of 4 096 blocks are employed. Two LZ-based algorithms, *LZMA* and *Zstandard* are also included, as well as the *bzip2* algorithm, based on the Burrows-Wheeler transform. These three algorithms are configured to obtain the best compression ratios. For completeness, results for an *arithmetic coder* and a *Huffman* coder are included in the comparison.

The average compression ratio for all packets sequences is computed for all the aforementioned algorithms. Compression time for each algorithm and packet sequence is measured as the minimum over 10 single-thread repetitions on a dedicated 3.20 GHz Intel Core i7-8700 CPU with 2 667 MT/s DDR4 memory. The average compression ratio and compression time per input byte are shown in Fig. 8. The CCSDS 124.0-B-1 algorithm produces compression ratios only surpassed by bzip2, LZMA and Zstandard. Of these, only bzip2 exhibits competitive compression speeds that improve upon CCSDS 124.0-B-1. However, note that bzip2 compresses all packets in each sequence together, which prevents real-time packet transmission and significantly increases memory requirements. Furthermore, CCSDS 124.0-B-1 is the only approach providing resiliency mechanisms against data loss events.

V. CONCLUSION

The CCSDS 124.0-B-1 standard is designed to achieve competitive coding efficiency for housekeeping telemetry data, while offering resiliency to data loss events and real-time transmission capabilities. Thanks to the proposed resynchronization strategies, the probability of being able to reconstruct a packet after a data loss event is increased, both immediately upon reception, and after having received subsequent packets. Users may adjust the desired degree of resiliency based on the expected error rates and reliability constraints of each mission phase. The proposed resynchronization strategies enhance the flexibility with which resiliency capabilities can be selected. Empirical evaluation using real and simulated data suggests that periodic activation of the new mask flag p_t every 5 to 10 packets yields the best coding performance, depending on the test corpus. Moreover, very smooth trade-offs between coding efficiency and resiliency can be used to assist selection. Upon joint analysis of coding efficiency and speed, CCSDS 124.0-B-1 yields the best trade-off, delivering highly competitive results compared to other algorithms. Results are only improved upon by bzip2, but bzip2 does not provide either data loss resiliency, or real-time compression, or a very small memory footprint, all of which are critical for processing remote sensing housekeeping telemetry.

APPENDIX. TABLE OF SYMBOLS

Symbols defined in the CCSDS 124.0-B-1 standard, the proposed resynchronization strategies and the experimental evaluation are listed in Table 3. The section of this document where they are introduced is also provided for each of them.

TABLE 3. Table of symbols.

Symbol	Name	Section
t	Time index	III-A
$\mathbf{I}_t$	Input packet	III-A
$\mathbf{M}_t$	Mask vector	III-A
$\mathbf{B}_t$	Build mask vector	III-A
$\mathbf{D}_t$	Change vector	III-B1
$\mathbf{X}_t$	Accumulated change vector	III-B1
R_t	Minimum required effective robustness level	III-B1
V_t	Effective robustness level	III-B1
C_t	Additional robustness level	III-B1
p_t	New mask flag	III-A
f_t	Full mask flag	III-B2
r_t	Uncompressed flag	III-B2
s_t^M	Synchronized mask flag	III-C1
s_t^I	Synchronized packet flag	III-C1
L_t^M	Unsynchronized mask counter	III-C1
L_t^I	Unsynchronized packet counter	III-C1
$T_{\text{new mask}}$	New mask period	IV-B
$T_{\text{full reference}}$	Full reference period	IV-D
$T_{\text{uncompressed}}$	Uncompressed period	IV-D
$T_{\text{full mask}}$	Full mask period	IV-D

ACKNOWLEDGMENT

Miguel Hernández-Cabronero would like to thank A. E. for their expert opinion and continued support.

REFERENCES

[1] D. Evans, J. A. Martinez-Heras, and R. Timm, "Housekeeping data: Can you afford not to compress it?" in *Proc. SpaceOps Conf. Delivering Dream Hosted NASA Marshall Space Flight Center Organized AIAA*, 2010, p. 2208.

[2] S. S. Saha, S. Rahman, M. U. Ahmed, and S. K. Aditya, "Ensuring cybersecure telemetry and telecommand in small satellites: Recent trends and empirical propositions," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 8, pp. 34–49, Aug. 2019.

[3] J.-A. Martínez-Heras, D. Evans, and R. Timm, "Housekeeping telemetry compression: When, how and why bother?" in *Proc. 1st Int. Conf. Adv. Satell. Space Commun.*, 2009, pp. 35–40.

[4] X. Shi, Y. Shen, Y. Wang, and L. Bai, "Differential-clustering compression algorithm for real-time aerospace telemetry data," *IEEE Access*, vol. 6, pp. 57425–57433, 2018.

[5] L. Guojun, S. Jian, and Z. Running, "Lossless data compression algorithm for satellite packet telemetry data," in *Proc. Int. Conf. Mech. Sci., Electric Eng. Comput. (MEC)*, Dec. 2013, pp. 2756–2759.

[6] P. Staudinger, J. Hershey, M. Grabb, N. Joshi, F. Ross, and T. Nowak, "Lossless compression for archiving satellite telemetry data," in *Proc. IEEE Aerosp. Conference.*, vol. 2, 2000, pp. 299–304.

[7] D. Evans and U. Moschini, "Ten times more information in your real-time TM," in *Proc. SpaceOps Conf.*, 2012, Art. no. 1275117.

[8] A. Liaghati and B. Meyer, "An adaptive DPCM method for efficient data compression in aerospace sensor systems," in *Proc. IEEE Aerosp. Conf.*, Mar. 2016, pp. 1–7.

[9] E. Paolini, M. Chiani, and G. P. Calzolari, "Long erasure correcting codes: The new frontier for zero loss in space applications?" in *Proc. SpaceOps Conf.*, Jun. 2006, p. 5827.

[10] D. J. Evans and A. Donati, "The ESA POCKET+ housekeeping telemetry compression algorithm: Why make spacecraft operations harder than it already is?" in *Proc. SpaceOps Conf.*, May 2018, p. 2613.

[11] *Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications*, Standard CCSDS 131.2-B-1, Consultative Committee for Space Data Syst. (CCSDS), Mar. 2012. [Online]. Available: https://public.ccsds.org/Pubs/131x2b1e1.pdf

[12] Y. Cho, H. Shin, and K. Kang, "Scalable coding and prioritized transmission of ECG for low-latency cardiac monitoring over cellular M2M networks," *IEEE Access*, vol. 6, pp. 8189–8200, 2018.

[13] H. Zhang, W. Bao, M. Yang, X. Li, and Y. Liu, "Adaptive classification fountain codes for reentry communication," *IEEE Access*, vol. 7, pp. 62911–62919, 2019.

[14] G. Quintarelli, M. Bertolucci, and P. Nannipieri, "Design and implementation of a DVB-S2 reconfigurable datapath BCH encoder for high data-rate payload data telemetry," *IEEE Access*, vol. 11, pp. 120281–120291, 2023.

[15] R. Schiavone, R. Garello, and G. Liva, "Performance improvement of space missions using convolutional codes by CRC-aided list Viterbi algorithms," *IEEE Access*, vol. 11, pp. 55925–55937, 2023.

[16] *Robust Compression of Fixed-Length Housekeeping Data*, Standard CCSDS 124.0-B-1, Consultative Committee for Space Data Syst. (CCSDS), 2023. [Online]. Available: https://public.ccsds.org/Pubs/124x0b1.pdf

[17] Y. He, X. Shi, and Y. Wang, "A modified LZW algorithm based on a character string parallel search in cluster-based telemetry data compression," *Electronics*, vol. 11, no. 17, p. 2656, Aug. 2022.

[18] S.-H. Hwang, K.-M. Kim, S. Kim, and J. W. Kwak, "Lossless data compression for time-series sensor data based on dynamic bit packing," *Sensors*, vol. 23, no. 20, p. 8575, Oct. 2023.

[19] H. Zhao, K. Song, K. Li, C. Wu, and Z. Chen, "A high-speed well logging telemetry system based on low-power FPGA," *IEEE Access*, vol. 9, pp. 8178–8191, 2021.

[20] P. Wan, Y. Zhan, and W. Jiang, "Study on the satellite telemetry data classification based on self-learning," *IEEE Access*, vol. 8, pp. 2656–2669, 2020.

[21] A. F. Shehab, M. A. Elshafey, and T. A. Mahmoud, "Recurrent neural network based prediction to enhance satellite telemetry compression," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–11.

[22] T. A. Mahmoud, A. F. Shehab, and M. A. Elshafey, "Different long short-term memory approaches to enhance prediction-based satellite telemetry compression," *J. Aerosp. Inf. Syst.*, vol. 18, no. 2, pp. 50–57, Feb. 2021.

[23] G. Ciaparrone, V. Benedetto, and F. Gissi, "A preliminary study on AI for telemetry data compression," in *Key Digital Trends in Artificial Intelligence and Robotics*, L. Troiano, A. Vaccaro, N. Kesswani, I. Díaz Rodriguez, I. Brigui, and D. Pastor-Escuredo, Eds. Cham, Switzerland: Springer, 2023, pp. 134–143.

[24] J.-G. MeßY, R. Schmidt, G. Fey, and F. Dannemann, "On the compression of spacecraft housekeeping data using discrete cosine transforms," in *Proc. Int. Workshop Tracking, Telemetry Command Syst. Space Appl. (TTC)*, 2016, pp. 1–8.

[25] J.-G. MeßY, R. Schmidt, and G. Fey, "Adaptive compression schemes for housekeeping data," in *Proc. IEEE Aerosp. Conf.*, 2017, pp. 1–12.

[26] S. Avramenko, S. Esposito, and M. Violante, "Rice coding-based lite compression algorithm with controlled error for sensor data," in *Proc. 5th IEEE Int. Workshop Metrology Aerosp. (MetroAeroSpace)*, Jun. 2018, pp. 6–10.

[27] *Reference Implementations of the CCSDS 124.0-B-1 Standard*. Accessed: Mar. 1, 2024. [Online]. Available: https://cwe.ccsds.org/sls/docs/SLS-DC/124.0-B-Info/Software/124BAvailabeSWImplementationsJune 2023.docx

[28] D. L. Wu, J. R. Piepmeier, J. Esper, N. Ehsan, P. E. Racette, T. E. Johnson, B. S. Abresch, and E. Bryerton, "IceCube: Submm-wave technology development for future science on a CubeSat," in *The Nanosatellite Revolution: 25 Years and Counting*. Bellingham, WA, USA: SPIE, 2019.

[29] J. Lambin, R. Morrow, L.-L. Fu, J. K. Willis, H. Bonekamp, J. Lillibridge, J. Perbos, G. Zaouche, P. Vaze, W. Bannoura, F. Parisot, E. Thouvenot, S. Coutin-Faye, E. Lindstrom, and M. Mignogno, "The OSTM/Jason-2 mission," *Mar. Geodesy*, vol. 33, no. sup1, pp. 4–25, Aug. 2010.

[30] H. Svedhem, D. Titov, D. McCoy, J.-P. Lebreton, S. Barabash, J.-L. Bertaux, P. Drossart, V. Formisano, B. Häusler, and O. Korablev, "Venus express—The first European mission to Venus," *Planet. Space Sci.*, vol. 55, no. 12, pp. 1636–1652, 2007.

[31] CCSDS Data Compression Working Group. *CCSDS 124.0-B-1 Test Corpora*. Accessed: Mar. 1, 2024. [Online]. Available: https://cwe.ccsds.org/sls/docs/SLS-DC/124.0-B-Info/TestData/ccsds124x0b1_test_dataset.zip

[32] *Lossless Data Compression*, Standard CCSDS 121.0-B-3, Aug. 2020. [Online]. Available: https://public.ccsds.org/Pubs/121x0b3.pdf

MIGUEL HERNÁNDEZ-CABRONERO received the B.Sc. degree in computer science and in mathematics from Universidad Autónoma de Madrid, Madrid, Spain, in 2010, and the M.Sc. and Ph.D. degrees in computer science from Universitat Autónoma de Barcelona (UAB), Spain, in 2011 and 2015, respectively. Since 2016, he has been collaborating with the Data Compression Working Group of the Consultative Committee for Space Data Systems. Since 2023, he has been an Associate Professor with UAB.

DAVID EVANS started work at ESA/ESOC gaining a broad experience on classic ESA missions, such as ERS-2 and CLUSTER. Then moved into the commercial area while working for EUTELSAT for ten years. He gained experience in cost control while managing the operation execution services for the fleet as it underwent a spectacular expansion from 5 to 19 satellites. Then moved into mission design, working for ESA/ESOC on low cost and fleet operations concept, where he co-initiated the OPS-SAT project (ESA's first cubesat).

JOAN BARTRINA-RAPESTA received the B.Sc., B.E., M.S., and Ph.D. degrees in computer science from UAB, in 2002, 2004, 2006, and 2009, respectively. Since 2012, he has been an Associate Professor with the Department of Information and Communications Engineering, UAB. His research interests include image coding topics, computing, and transmission.

FRANCESC AULÍ-LLINÀS (Senior Member, IEEE) received the B.E. (Hons.) and Ph.D. (cum laude) degrees in computer science from UAB, in 2002 and 2006, respectively. He is currently an Associate Professor with the Department of Information and Communications Engineering, UAB. His research interests include image and video coding, computing, and transmission. In 2013, he received a Distinguished R-Letter given by the IEEE Communications Society for a paper coauthored with Michael Marcellin.

IAN BLANES received the B.S., M.S., and Ph.D. degrees in computer science from UAB, in 2007, 2008, and 2010, respectively. Since 2011, he has been actively involved in the creation of new on-board data compression standards within the framework of the CCSDS. Since 2003, he has also been with the Group on Interactive Coding of Images, UAB, where he is currently an Associate Professor. His research interest includes data compression in spaceborne instruments.

JOAN SERRA-SAGRISTÀ (Senior Member, IEEE) received the Ph.D. degree in computer science from UAB, in 1999. From 1997 to 1998, he was with the University of Bonn, Bonn, Germany, funded by DAAD. He is currently a Full Professor with the Department of Information and Communications Engineering, UAB. He has coauthored more than 100 articles. His research interest includes data compression, especially image coding for remote sensing applications.