

This is the **accepted version** of the journal article:

Bartrina Rapesta, Joan; Blanes Garcia, Ian; Aulí Llinàs, Francesc; [et al.]. «A Lightweight Contextual Arithmetic Coder for On-Board Remote Sensing Data Compression». IEEE Transactions on Geoscience and Remote Sensing, Vol 55, Issue 8 (August 2017), p. 4825-4835. DOI 10.1109/TGRS.2017.2701837

This version is available at <https://ddd.uab.cat/record/299984>

under the terms of the  **IN
COPYRIGHT** license

A Lightweight Contextual Arithmetic Coder for On-Board Remote Sensing Data Compression

Joan Bartrina-Rapesta, Ian Blanes, Francesc Aulí-Llinàs, Joan Serra-Sagristà, Victor Sanchez and Michael W. Marcellin

Abstract—The Consultative Committee for Space Data Systems (CCSDS) has issued several data compression standards devised to reduce the amount of data transmitted from satellites to ground stations. This paper introduces a contextual arithmetic encoder for on-board data compression. The proposed arithmetic encoder checks, at most, the causal adjacent neighbours to form the context and uses only bitwise operations to estimate the related probabilities. As a result, the encoder consumes few computational resources, making it suitable for on-board operation. Our coding approach is based on the prediction and mapping stages of CCSDS-123 lossless compression standard, an optional quantizer stage to yield lossless or near-lossless compression, and our proposed arithmetic encoder. For both lossless and near-lossless compression, the achieved coding performance is superior to that of CCSDS-123, M-CALIC and JPEG-LS. Taking into account only the entropy encoders, FLW is slightly better than MQ and Interleaved Entropy Coding.

Index Terms—Remote Sensing Data Compression, CCSDS-123, Lossless and Near-Lossless Coding, Arithmetic Coding.

I. INTRODUCTION

REMOTE sensing imagery is becoming an invaluable tool for governments, rescue teams and aid organizations to manage infrastructure and natural resources, to appraise climate changes, or to give support when natural disasters strike. Since remote sensing images tend to be very large, high-performance compression techniques are of paramount importance.

Let I , J , and K , be the number of columns, rows, and components of an image x , and let $x_{i,j,k}$ denote a pixel at location (i, j, k) of the image. Such an image is commonly compressed employing one of three regimes: lossless compression, which allows perfect reconstruction of the original image x ; lossy compression, which approximates x , introducing an error in the reconstructed image x' that enables a higher compression ratio than possible with lossless compression; or near-lossless compression, which is a particular case of lossy compression where the peak absolute error (PAE) of x' is controlled during the coding process with a tolerance value Δ . Specifically,

$$\max_{i,j,k} \{|x_{i,j,k} - x'_{i,j,k}|\} \leq \Delta. \quad (1)$$

This work was supported in part by the Spanish Ministry of Economy and Competitiveness (MINECO) and by the European Regional Development Fund (FEDER) under Grant TIN2015-71126-R, by the Catalan Government under Grant 2014SGR-691, and by Centre National d'Études Spatiales (CNES).

J. Bartrina-Rapesta, I. Blanes, F. Aulí-Llinàs, and J. Serra-Sagristà are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, E-08193 Cerdanyola del Vallès, Barcelona, Spain (e-mail: joan.bartrina@uab.cat).

V. Sanchez is with University of Warwick, CV4 7AL, Coventry, UK.

M.W. Marcellin is with University of Arizona, Tucson, AZ 85721-0104, USA.

Within the Consultative Committee for Space Data Systems (CCSDS) [1], the Multispectral & Hyperspectral Data Compression Working Group is in charge of proposing techniques for remote sensing data compression. Such techniques are mainly developed to be implemented on board, where limited resources are available and low complexity encoders are needed. In 1997, CCSDS published CCSDS-121.0-B-1 [2], aimed at lossless data compression. In 2005, CCSDS published CCSDS-122.0-B-1 [3], devised for lossless and lossy compression of monocomponent images based on wavelet transforms. In 2012, CCSDS published its latest standard, CCSDS-123.0-B-1 [4], focused on lossless compression for multispectral and hyperspectral images based on prediction. Note that, to date, there is no CCSDS standard proposal devised to multispectral and hyperspectral images for near-lossless coding. In what follows, we will refer to CCSDS-123.0-B-1 as CCSDS-123.

Lossless and near-lossless coding is an active research topic, as witnessed by the amount of recent publications in the last decade [5]–[17]. Some of these contributions, such as [7], [11], [15]–[17], yield better coding performance than CCSDS-123 for lossless compression but at the expense of an increased computational complexity. Among them, results provided in [7] can be misleading, since they were obtained using images from the 1997 AVIRIS products, which are known to have undergone an inappropriate calibration [18]. Next three contributions [11], [15], [16] yield better coding performance than CCSDS-123, but at the expense of an increased computational complexity due to the expensive algorithms applied to improve prediction estimation. The last contribution [17] yields competitive coding performance by including a light spectral regression in the spectral domain, which has a low computational cost.

It is worth noting that none of the previous techniques provides support for near-lossless coding, which is demanded if even better coding performance is requested. Near-lossless coding [5], [6], [8]–[10], [12]–[14] can yield higher compression ratios at a bounded distortion of $\Delta > 0$. Some of the most prominent recent contributions for near-lossless compression are: [12], which presents an overview of the latest coding standards for remote sensing, including a near-lossless version of CCSDS-123; [8], [9], which introduce a near-lossless coding based on wavelet transforms; [14], which goes one step further, proposing an embedded near-lossless coding system based on wavelet transform and prediction coding; and [13], which presents a rate-control method for predictive image encoders using the CCSDS-123 predictor. Most of the latest contributions use the CCSDS-123 predictor, since it is suitable for being used on-board thanks to its low

complexity and high decorrelation efficiency.

After the predictor of CCSDS-123 one can choose between a sample- or a block-adaptive encoder. The sample-adaptive encoder achieves better performance than the block-adaptive encoder when the signal is encoded at more than 1 bits per sample (bps). However, because the minimum codeword length of the sample-adaptive encoder is one bit, block-adaptive encoding yields superior performance for signals that can be encoded at less than 1 bps.

Although context-based arithmetic encoders typically obtain excellent coding performance at all rates, they are not included in CCSDS-123 because they can have a high computational demand owing to: a) probability estimation, b) the renormalization procedure, and c) context formation, which are expensive operations and are executed intensively. Despite the computational demand of context-based arithmetic encoders, they are included in some remote sensing coding approaches [6], [19], [20]. Contributions aimed to reduce the computational load by estimating the probability using multiplication-free implementations can be found in the literature: the Q coder [21] approached the interval division by means of lookup tables; and the M coder [22] uses a reduced range of possible subinterval sizes together with look up tables. Some methods based on these approaches have been introduced in different standards [23]–[26]. The operations carried out by the renormalization procedure can be avoided if, instead of producing a single codeword, the coder produces short codewords of fixed length [27], [28]. In particular, [28] presents a context-adaptive binary arithmetic coder with fixed-length codewords (FLW) that outperforms the MQ [29] and M coders in terms of coding performance. FLW avoids the renormalization procedure but still estimates probabilities through the division.

It is worth noting that none of the previously mentioned contributions is devised to reduce the computation related with probability estimation and the renormalization simultaneously. In this work, we propose an arithmetic encoder that: 1) utilizes inexpensive operations to estimate probabilities, 2) does not incorporate the renormalization procedure, and 3) employs a simple context model. It yields strong coding performance at low and high rates for remote sensing images. Our probability estimation procedure builds on that of FLW. Originally, FLW uses a sliding window to estimate the probability of the symbols coded using a division operation. Herein, the sliding window size of FLW is adapted to deal only with power of two sizes, which allows the use of low complexity bitwise operations and spares the division.

The proposed arithmetic coder is incorporated in a lossless and near-lossless coding scheme, providing improved compression performance over current remote sensing image compression techniques. Roughly described, the adopted coding scheme departs from the predictor and mapping included in CCSDS-123 and i) utilizes a near-lossless quantizer, ii) employs a binary arithmetic coder that operates on a line-by-line and bitplane-by-bitplane basis, iii) introduces a new context model that evaluates (at most) only causal adjacent samples, and iv) uses only bitwise operations to estimate symbol probabilities. Extensive experimental results indicate that

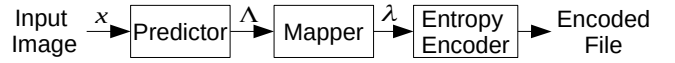


Fig. 1: CCSDS-123 encoding scheme.

our proposed approach improves upon CCSDS-123 in terms of lossless compression ratios, and also outperforms a near-lossless version of the sample-adaptive and block-adaptive coder of CCSDS-123, JPEG-LS [30] and M-CALIC [6] in terms of lossless and near-lossless coding performance. Comparing only the entropy encoders, FLW is slightly better than MQ and Interleaved Entropy Coder (IEC) [31].

The rest of the paper is structured as follows: Section II briefly reviews the CCSDS-123 coding system and a near-lossless technique for coding systems based on prediction, Section III describes our proposed context-based arithmetic coder with bitwise probability estimation, Section IV describes how our proposed arithmetic coder is incorporated in a coding scheme that uses the predictor of CCSDS-123, Section V presents the experimental results, and Section VI concludes this work.

II. CCSDS-123 AND NEAR-LOSSLESS COMPRESSION

A. CCSDS-123

The CCSDS-123 standard, which is limited to encoding samples of $N = 16$ bits per pixel per band, can be structured in three stages: *Predictor*, *Mapper* and *Entropy Encoder*. Fig. 1 illustrates the encoding pipeline of CCSDS-123.

In summary, the predictor estimates the value of the current sample $x_{i,j,k}$ using previously scanned samples. This predicted sample is denoted by $\tilde{x}_{i,j,k}$. The prediction error Λ is computed as

$$\Lambda_{i,j,k} = x_{i,j,k} - \tilde{x}_{i,j,k}, \quad (2)$$

and then mapped to a non-negative integer $\lambda_{i,j,k}$ called the mapped prediction residual. The entropy encoder, is in charge of encoding $\lambda_{i,j,k}$ without loss. For the entropy encoder in CCSDS-123, one can choose between a sample- or a block-adaptive encoder.

Further details of the CCSDS-123 stages can be found in [12] and [32].

B. Near-Lossless Compression

For the encoder described above, the decoder can reproduce $x_{i,j,k}$, without loss. In this section, we discuss the addition of a quantizer, which results in higher compression ratios, but at the expense of some loss of fidelity in the decompressed image. The simplest and most effective way to design a prediction-based lossy compression algorithm is to quantize the prediction error $\Lambda_{i,j,k}$ with a quantizer Q , resulting in quantized-then-dequantized version $\hat{\Lambda}_{i,j,k}$ (and, in consequence, $\hat{\lambda}_{i,j,k}$). The resulting quantization index is referred to as $\Lambda_{i,j,k}^Q$, and its remapped version is denoted by $\lambda_{i,j,k}^Q$. Subsequent predictions $\tilde{x}_{i,j,k}$ are calculated using previous reconstructed (lossy) samples $\hat{x}_{i,j,k}$, which are obtained by implementing

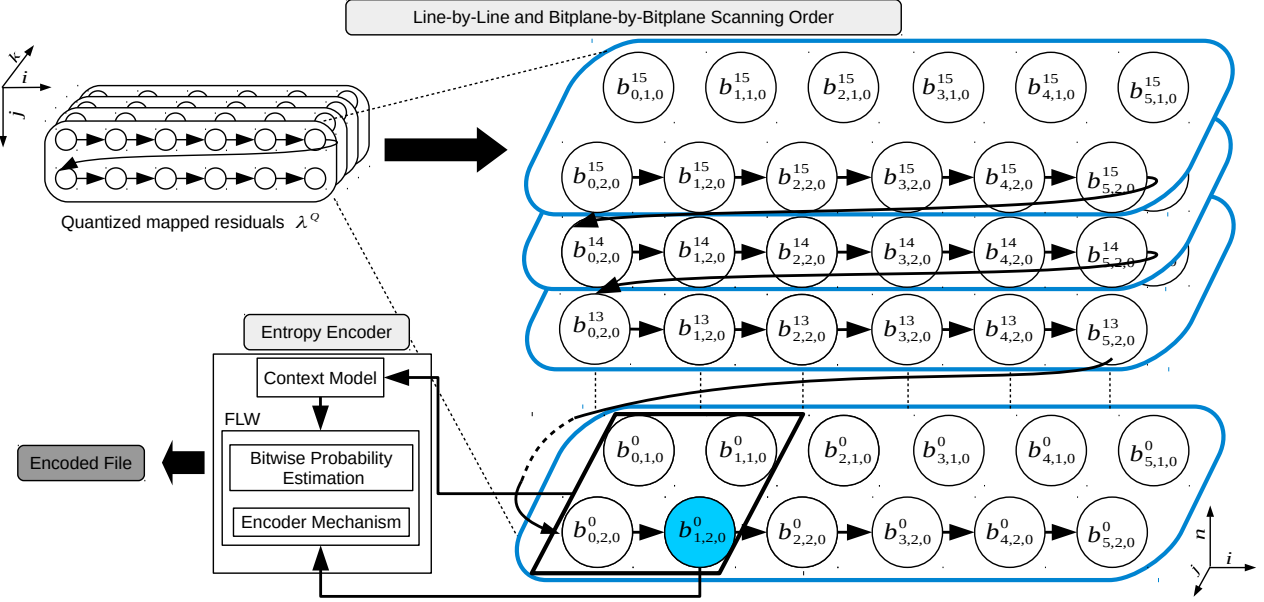


Fig. 2: Illustration of the scanning order and the entropy encoder.

a decoder in the encoder [12], [33]. The decoder creates the reconstructed (lossy) image samples via:

$$\hat{x}_{i,j,k} = \hat{\Lambda}_{i,j,k} + \tilde{x}_{i,j,k}. \quad (3)$$

It is worth noting that the errors in the reconstructed pixels are identical to the errors introduced in the prediction errors by the quantizer. That is, $x_{i,j,k} - \hat{x}_{i,j,k} = \Lambda_{i,j,k} - \hat{\Lambda}_{i,j,k}$. Thus, the errors in reconstructed pixels can be precisely controlled by controlling the individual quantization errors. This is the basis of “near-lossless compression.”

III. A LIGHTWEIGHT BINARY ARITHMETIC CODER WITH CONTEXT MODEL

The entropy encoder presented in this paper works with binary symbols. To this end, we denote the n -th bit of the binary representation of $\lambda_{i,j,k}^Q$ by $b_{i,j,k}^n$, with $N-1 \geq n \geq 0$. Here, N is chosen to provide a sufficient number of bits to represent all the $\lambda_{i,j,k}^Q$, being $b_{i,j,k}^{N-1}$ the most significant bit.

To facilitate use with on-board sensors, our proposal processes data in a line-by-line fashion. Once a line is scanned, predicted, and mapped to positive values, it is entropy encoded on a bitplane-by-bitplane basis. The entropy encoder makes use of context model patterns obtained using a context window that contains symbols coded previously to the current symbol. The top left portion of Fig. 2 displays the quantized and remapped prediction residuals λ^Q . The binary representation of these samples is shown on the right, while the bottom left portrays the entropy encoder, which is fed by the current bit to be encoded and its context. The bit to be encoded is shaded in blue, while the context window is framed with a rectangle.

A. Context Model

Let \mathbf{M} be the set of all possible patterns that can occur within the context window, with context $m \in \mathbf{M}$ being a particular realization, resulting in a context index $c \in \mathbf{C} =$

$\{0, \dots, C-1\}$. These context indices (loosely referred to as contexts in what follows) are determined by a modeling function $F: \mathbf{M} \rightarrow \mathbf{C}$. For each bit b to be coded, a probability model is used, corresponding to its context c . In particular, the probability model estimates the conditional probability $p(b|c) = p(b|F(m))$. After encoding, the probability model is updated with the latest coded bit b . That is, $p(b|c)$ is estimated on the fly. Specifically, our probability model estimates the probability $p(b=0|c)$. A careful design of the context model is required to obtain high coding efficiency. This task is complicated by the goal of achieving low encoder complexity for the purpose of operating on onboard remote sensing scenarios.

A simple strategy for context modeling employs a context window that contains only the three nearest causal neighbors as depicted in Fig. 2. We consider several choices for the context modeling function F . The first ignores all samples within the context window except the one directly above the sample of interest. This is indicated in Fig. 3 (a). Three other choices are shown in Fig. 3 (b), (c) and (d). The notation V, H, HV, and HVD is used in the figure, where V (Vertical) denotes the sample above the bit to be encoded, H (Horizontal) denotes the sample to the left, and D (Diagonal) denotes the sample to the left and above. To take advantage of dependencies between spectral components, the preceding spectral component $k-1$ can be included in the context window. In this case, S (Spectral) denotes the co-registered sample in the previous spectral component. The inclusion of this sample gives rise to five additional modeling functions as shown in parts (e) through (i) of the figure. Note that if only samples H and S are employed by the modeling function, only the current scanned line must be stored in memory. For all other modeling functions, the previous and the current lines are necessary.

Rather than the actual bit (from bitplane n) of each neighboring sample, the so-called “significance state” is employed

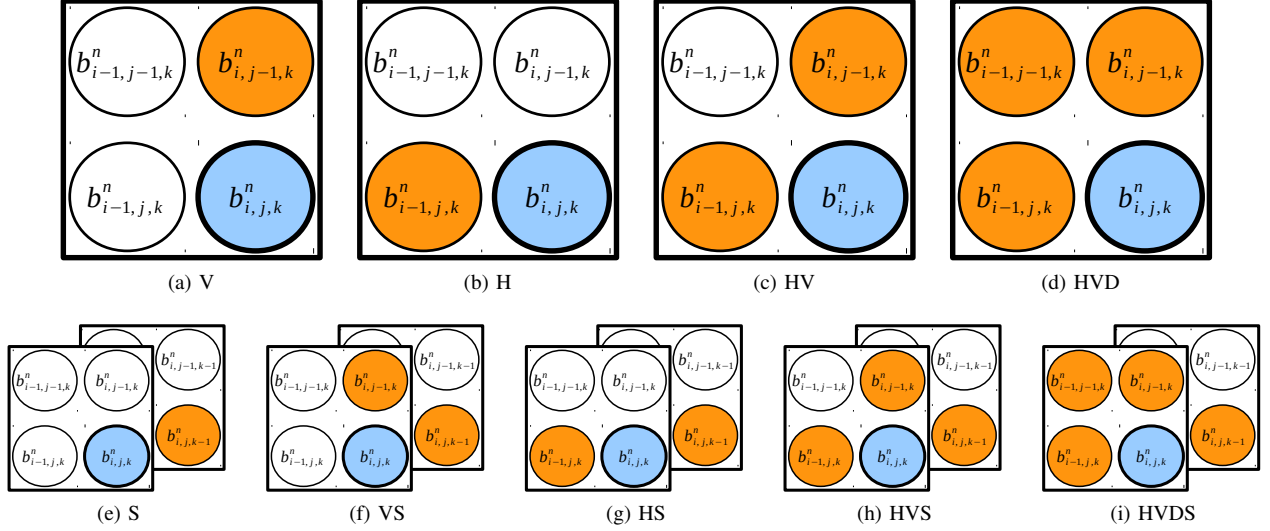


Fig. 3: Illustration of different context models to encode $b_{i,j,k}^n$.

TABLE I: Context assignments for the V, H, HV, and HVD modeling functions.

c	V	H	HV		HVD		
	$s_{i,j-1,k}^n$	$s_{i-1,j,k}^n$	$s_{i,j-1,k}^n$	$s_{i-1,j,k}^n$	$s_{i,j-1,k}^n$	$s_{i-1,j,k}^n$	$s_{i-1,j-1,k}^n$
0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1
2			1	0	0	1	0
3			1	1	0	1	1
4					1	0	0
5					1	0	1
6					1	1	0
7					1	1	1

to compute the context c . To this end, let $s_{i,j,k}^n$ denote the significance state of the sample at location i, j, k at bitplane n . A value of 1 indicates that the sample contains a 1 at bitplane n or higher. Table I shows how c is derived from the significance states of the neighbors for the V, H, HV, and HVD modeling functions. The S modeling function results in two states, i.e., $c \in \{0, 1\}$. The VS, HS, HVS, and HVDS modeling functions result in twice the number of states than their counterparts that do not employ S. They are not shown in the table for the sake of space. Experimental results for all context modeling functions are presented in a subsequent section.

Before finishing this section, we note that the entropy coder and its associated probability models are initialized at the beginning of each bit plane of each component. In particular, the initial probability model for each context is set to a value of $p(b = 0|c) = 0.66$. The probability is biased towards 0 since, as found empirically, bits of higher bitplanes have higher probability of being 0, thus allowing FLW to adapt faster. This, together with the fact that all bitplane data from the current line (and its predecessor, when relevant) are available in the encoder, leads to the conclusion that the bitplanes of the current line can be encoded in parallel. This parallel strategy is not possible in the decoder. The use of significant states in the context formation process requires that bitplanes be decoded sequentially. We note that the probabilities are reset ($p(b = 0|c) = 0.66$) at the beginning of each component with-

out penalizing the coding performance. This is because only 2^{12} symbols are encoded with the default probability value, which on average for the image corpora used, corresponds to the 0.06% of the total symbols per band to be encoded.

B. Bitwise Probability Estimation

As mentioned before, FLW was devised to reduce computational costs through the use of fixed length codewords, which avoids a renormalization operation, but is not aimed to reduce the computational load derived from probability estimation [28]. For each context c , FLW uses a sliding window of symbols coded with that context. The length of this window varies between \mathcal{T} and $2\mathcal{T}-1$ symbols. The probability estimate is updated once every \mathcal{V} symbols coded, according to

$$p(b = 0|c) = \frac{Z \ll \mathcal{B}}{W}, \quad (4)$$

with W representing the number of symbols within the window, Z the number of zeroes within the window, and \mathcal{B} the number of bits used to express symbol probabilities. The numerator of the expression is computed by left-shifting the binary representation of Z by \mathcal{B} bits. The size of the window is incremented each time a symbol is encoded using context c until $W = 2\mathcal{T}$, at which time the window size is immediately

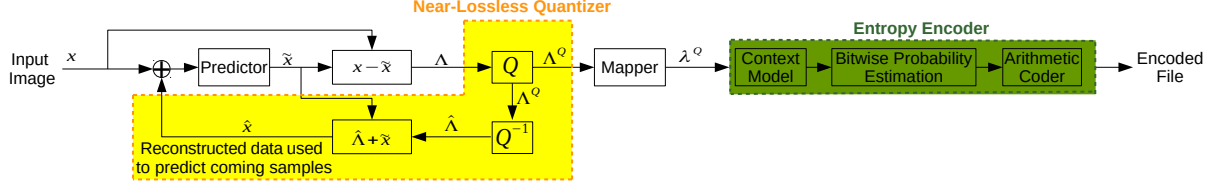


Fig. 4: Adopted coding approach.

	Predictor	Quantizer	Mapper	Data Scanned	Supported	Context Model	Entropy Encoder
CCSDS-123	✓	✗	✓	Line or Block		✗	Sample-Adaptive or Block-Adaptive
Proposal	✓	✓	✓	Line		✓	Contextual Binary Arithmetic Coder

TABLE II: Main differences between CCSDS-123 and the adopted approach.

reduced to \mathcal{T} and the number of zeroes within the window is updated according to $Z \leftarrow Z - Z'$, and $Z' \leftarrow Z$, with Z' being the number of zeroes coded during the most recent \mathcal{T} symbols.

In the original approach of FLW as formulated above, $p(b|c)$ is computed via a division operation to achieve maximum accuracy. Such a division may tax the on-board computational resources in a remote sensing scenario. To reduce computational complexity, we propose to estimate the probability through bitwise operations. The substitution of the division by bitwise operations requires that $\mathcal{V} = \mathcal{T}$ and that both are a power of two. This forces the sliding window to contain a power of two symbols, so the probability can be updated using only bit shift operations according to

$$p(b|c) = (Z \ll \mathcal{B}) \gg \log_2(W). \quad (5)$$

W and Z are then updated through $W \leftarrow W \gg 1$ and $Z \leftarrow Z \gg 1$. Note that this update rule for Z only approximates the number of zeroes in the most recent \mathcal{T} coded samples. Nevertheless, the update can be carried out in the decoder using the same approximation. At the beginning of encoding, the probability is first updated when \mathcal{V} symbols are coded. Subsequently, it is updated every $\mathcal{V}/2$ symbols. The strategy proposed here can be seen as a special case of (4), which was not explored in [28].

Using (5) instead of (4) reduces the flexibility of the arithmetic coder since the updating of the probability estimates and the window size are tied together. The maximum performance achieved with the original formulation of the arithmetic coder proposed in [28] is achieved when the probability estimate is updated every symbol, i.e., $\mathcal{V} = 1$, regardless of the window size. The strategy proposed here provides a significant reduction in complexity with a minor reduction in compression performance. Experimental results provided in Section V indicate that our approach yields highly competitive compression performance.

IV. ADOPTED CODING APPROACH

Although the novel entropy encoder presented here may be incorporated in any coding system, we employ it in the CCSDS-123 coding pipeline. Fig. 4 illustrates the adopted

coding approach, which employs the predictor and mapper of CCSDS-123, but adds a near-lossless quantizer (highlighted in yellow), and substitutes the usual CCSDS-123 encoder by our entropy encoder (highlighted in green). The circle containing a cross at the left side of the figure indicates that the input to the predictor is either the original pixel x (when the optional quantization is not present) or the reconstructed pixel \hat{x} (when quantization is present).

The adopted coding scheme is evaluated with a Uniform Quantizer (UQ) and a Uniform Scalar Deadzone Quantization (USDQ) [29]. The UQ operates over $\Lambda_{i,j,k}$ to obtain a quantization index according to

$$\Lambda_{i,j,k}^Q = \text{sign}(\Lambda_{i,j,k}) \left\lfloor \frac{|\Lambda_{i,j,k}| + \Delta}{2\Delta + 1} \right\rfloor, \quad (6)$$

where $2\Delta + 1$ is the quantization step size. The operation to reconstruct $\hat{\Lambda}_{i,j,k}$ from its quantization index is given by

$$\hat{\Lambda}_{i,j,k} = \text{sign}(\Lambda_{i,j,k}^Q)(2\Delta + 1)\Lambda_{i,j,k}^Q. \quad (7)$$

UQ is employed in lossless compression techniques such as JPEG-LS, M-CALIC and 3D-CALIC [34]. On the other hand, the USDQ quantizes $\Lambda_{i,j,k}$ to obtain a quantization index according to

$$\Lambda_{i,j,k}^Q = \text{sign}(\Lambda_{i,j,k}) \left\lfloor \frac{|\Lambda_{i,j,k}|}{\Delta + 1} \right\rfloor, \quad (8)$$

where the quantization step is $\Delta + 1$. The operation to reconstruct $\hat{\Lambda}_{i,j,k}$ from its quantization index is expressed as

$$\hat{\Lambda}_{i,j,k} = \text{sign}(\Lambda_{i,j,k}^Q)(\Delta + 1)\Lambda_{i,j,k}^Q. \quad (9)$$

Due to its straightforward implementation and excellent performance, USDQ has been selected for the JPEG 2000 standard [24]. USDQ partitions the range of input values into intervals all of size Δ , except for the interval that contains zero, which is of size 2Δ . This results in all absolute pixel errors $|x_{i,j,k} - \hat{x}_{i,j,k}|$ being bounded above Δ for both quantizers.

Table II summarizes the main differences between CCSDS-123 and the adopted coding scheme.

TABLE III: Summary of data used in the experimental results. Sensor name, its abbreviation, number of images from each sensor, and first-order entropies (in bps) on average per sensor are provided. The last two columns indicate the predictor mode and the local sum used for each sensor.

Sensor	Abbreviation	Number of images	Entropy	Predictor Mode	Local Sum
Aviris Calibrated	AC	5	9.77	Neighbor Oriented	Full Mode
Aviris Uncalibrated	AU	7	11.21	Neighbor Oriented	Full Mode
Airs	A	9	11.34	Neighbor Oriented	Reduced Mode
Casi	C	2	10.52	Neighbor Oriented	Reduced Mode
Crism	Cr	20	10.69	Column Oriented	Reduced Mode
Hyperion	H	4	9.53	Column Oriented	Reduced Mode
M3	M3	5	9.19	Column Oriented	Reduced Mode
Total / average	—	52	10.41	—	—

V. EXPERIMENTAL RESULTS

This section presents a set of experiments aimed at the analysis and evaluation of the adopted coding scheme. First the proposed context modeling functions are evaluated in terms of the conditional entropy of the prediction residual. The bitwise probability estimator is then evaluated via the same performance metric to determine its proper configuration. A variety of binary encoder mechanisms such as IEC, MQ and FLW are evaluated in terms of their lossless compression performance in conjunction with the proposed context modeling and probability estimation. Finally, the resulting proposed overall approach is compared in terms of lossless and near-lossless compression performance with CCSDS-123, JPEG-LS, and M-CALIC.

For the experiments conducted in this work, we have selected a set of images¹ collected with different sensors that are included in CCSDS MHDC-WG corpus. Sensor names and their main features are listed in Table III. The average entropy is reported for each image type. The reported values are first order entropy; they represent the entropy of individual pixels, without accounting for any dependencies among pixels within or between components.

In [35], the impact of different CCSDS-123 parameters that control the operation of the prediction and the entropy encoder were evaluated, suggesting that a correct parameter selection had more impact on the predictor stage than in the entropy encoder stage. Concerning the prediction, parameters local sum type, prediction mode, number of prediction bands, and predictor adaption rate were the most critical. Extensive experimental evaluations were conducted to find suitable configurations.

In the current manuscript, leaning on results in [35] and after conducting also an extensive evaluation, experimental results are produced for the following parameter configuration: local sum type and predictor mode depend on the acquisition sensor (as indicated in the last two columns of Table III); the number of prediction bands P is set to 3, since it is a good tradeoff between computational load and coding performance; the predictor adaptation rate ν_{max} is set to 3, since, in general,

it yields the best performance.

For evaluating the performance of the context modelling and probability estimation, we employ the conditional entropy of the prediction residuals, as mentioned above. For the work proposed here, binary entropy coding is employed. To yield results with units in bits per pixel the binary entropies of all bitplanes are added. Since our context model estimates the probability of $p(b = 0|c)$, the conditional entropy of an image (in bits) is computed as:

$$H(\lambda^Q) = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} \sum_{n=0}^{15} \begin{cases} \log_2(p(b_{i,j,k}^n = 0|c)) & \text{if } b_{i,j,k}^n = 0 \\ \log_2(1 - p(b_{i,j,k}^n = 0|c)) & \text{if } b_{i,j,k}^n = 1 \end{cases}, \quad (10)$$

where λ^Q denotes the symbols to be entropy coded.

A. Context Modeling Function

The context model is used to select the probability model that is employed to encode the current symbol. In this first experiment, each of the probability models themselves are estimated using the high performance method given by (4) employing $\mathcal{V} = 1$ and $\mathcal{T} = 2^{12}$, without regard to complexity.

Table IV provides the conditional entropy obtained (in bps) for the different context formations defined in Section III-A, i.e., V, H, HV, HVD, S, VS, HS, HVS, and HVDS. Results from the table suggest that: 1) all of the modeling functions provide significant improvements over the pixel entropy reported in Table III; 2) differences in performance between the modeling functions are generally small; 3) although the context models H and S yield the worst performance on average, they are the best option when memory resources are severely limited since they need only store samples from the current line to be encoded; 4) adding the S sample to a context results in an improvement of only about 0.01 bps; 5) the V context obtains a coding benefit of 0.02 bps on average with respect to the H context, and only adds the previous processed line to its storage requirements. In what follows, we select context model V for further evaluation due to its favorable trade off between performance, memory resources, and computational load.

¹The images used are available at <http://cwe.ccsds.org/sls/docs/sls-dc/123.0-B-Info/TestData>.

TABLE IV: Conditional entropy of the prediction residuals (in bps) for the context modeling functions denoted by V, H, HV, HVD, S, VS, HS, HVS, and HVDS. Results are reported on average for different sensors and $\Delta = 0$.

Sensor	Context Formation								
	V	H	HV	HVD	S	VS	HS	HVS	HVDS
AC	3.69	3.69	3.68	3.68	3.69	3.68	3.68	3.68	3.67
AU	5.01	5.00	5.00	4.99	5.00	4.99	4.99	4.99	4.99
A	4.24	4.23	4.24	4.24	4.24	4.24	4.24	4.24	4.23
C	4.85	4.85	4.84	4.84	4.85	4.84	4.84	4.83	4.83
Cr	4.15	4.21	4.14	4.14	4.20	4.12	4.19	4.12	4.12
H	4.26	4.26	4.25	4.25	4.29	4.26	4.26	4.25	4.25
M3	2.66	2.70	2.65	2.65	2.70	2.64	2.68	2.63	2.63
Average	4.12	4.14	4.11	4.11	4.14	4.11	4.13	4.10	4.10

TABLE V: Conditional entropy of the prediction residuals (in bps) for $\Delta = 0$ resulting from the Maximum Precision and the Bitwise probability estimators. The V context model is employed in each case. Best results for each strategy are enhanced in boldface.

$\mathcal{T} =$	division $\mathcal{V} = 1$					bitwise operations $\mathcal{V} = \mathcal{T}$				
	2^{16}	2^{14}	2^{12}	2^{10}	2^8	2^{16}	2^{14}	2^{12}	2^{10}	2^8
AC	3.70	3.69	3.69	3.70	3.75	3.72	3.70	3.69	3.70	3.77
AU	5.02	5.01	5.01	5.01	5.06	5.04	5.02	5.01	5.02	5.08
A	4.27	4.26	4.23	4.24	4.26	4.32	4.29	4.24	4.26	4.28
C	4.94	4.86	4.85	4.86	4.89	4.90	4.86	4.85	4.86	4.91
Cr	4.16	4.15	4.15	4.15	4.18	4.21	4.17	4.16	4.16	4.20
H	4.27	4.26	4.26	4.27	4.31	4.29	4.27	4.26	4.27	4.32
M3	2.69	2.66	2.66	2.67	2.70	2.71	2.68	2.67	2.68	2.72
Average	4.15	4.13	4.12	4.13	4.16	4.17	4.14	4.13	4.13	4.18

B. Probability Estimation

This subsection reports the results obtained by the two different probability estimation strategies discussed in Section III. In particular, Table V reports the conditional entropy of the prediction residuals resulting from the two different probability estimation strategies. In both cases, the V context model is employed. The left side of the table presents results for the Maximum Precision technique (using division) as defined by equation (4). These results are shown for different values of \mathcal{T} , but $\mathcal{V} = 1$. The right side of the table presents results for the Bitwise strategy as defined by equation (5). The same values of \mathcal{T} are explored, but always with $\mathcal{V} = \mathcal{T}$, as required to avoid division. The results suggest that $\mathcal{T} = 2^{12}$ attains the highest performance for both strategies. Larger \mathcal{T} degrades the coding performance because the window may contain symbols that are not correlated to the current one. Smaller \mathcal{T} degrades the coding performance because there are insufficient symbols to reliably estimate the probabilities $p(b|C)$. The results of Table V also indicate that the low complexity strategy that employs bitwise operations is as competitive as that employing division. Although not tabulated here for the sake of space, these results hold for the other context modeling functions considered in previous sections.

TABLE VI: Coding performance (in bps) of the proposed approach using MQ, IEC, and FLW entropy encoding. All results employ context model V and bitwise probability estimation with $\mathcal{T} = \mathcal{V} = 2^{12}$.

Sensor	MQ	IEC	FLW
AC	3.74	3.72	3.71
AU	5.06	5.04	5.03
A	4.30	4.29	4.28
C	4.90	4.88	4.87
Cr	4.20	4.18	4.18
H	4.31	4.29	4.28
M3	2.71	2.70	2.69
Average	4.17	4.16	4.15

C. Entropy Coding

We note that the context model and probability estimator proposed here can be used with any entropy encoder that codes binary symbols according to a given probability model, such as MQ, IEC or the adopted FLW. Table VI provides actual compression results (in bps) obtained using the MQ, IEC and FLW entropy coders. In each case, the results are obtained with context model V and the bitwise estimator with $\mathcal{T} = \mathcal{V} = 2^{12}$.

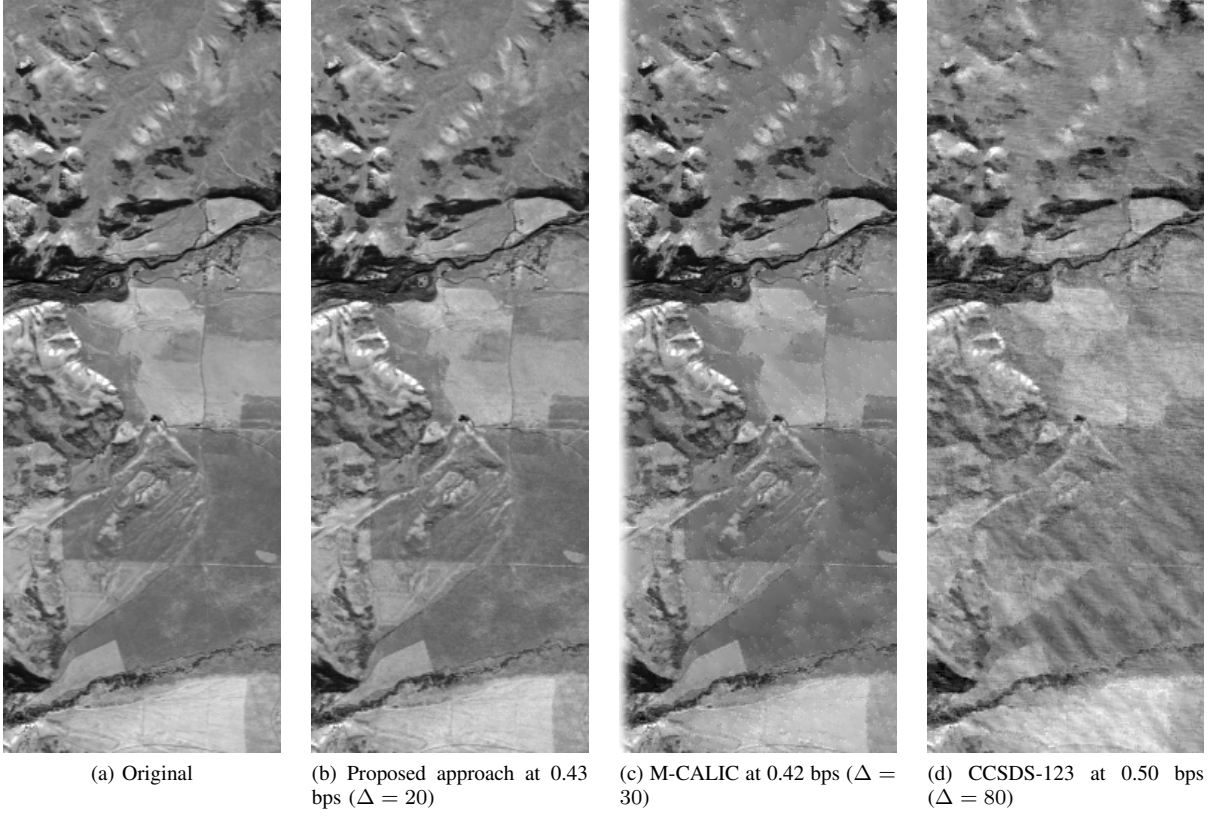


Fig. 5: Visual comparison for “Avisir Calibrated Yellowstone sc00” image.

From these results we can see that, on average, FLW yields slightly better results than IEC and MQ.

D. Lossless and Near-Lossless Compression

The results reported in this subsection compare the lossless performance of the proposed approach with JPEG-LS, M-CALIC and CCSDS-123. Additionally, we compare its *near-lossless* performance to that of JPEG-LS, M-CALIC and an implementation of CCSDS-123. Different quantizers have been combined with our proposal and CCSDS-123, to obtain an as fair as possible comparison. In particular, the uniform quantizer (UQ) and the dead-zone quantizer (USDQ) discussed in Section IV are compared.

M-CALIC and the near-lossless version of CCSDS-123 are considered to be state-of-the-art in terms of compression performance and computational complexity, and JPEG-LS is a standard technique with near-lossless features. All results for the proposed scheme are produced using the FLW arithmetic coder, context model V and the bitwise probability estimator having $\mathcal{V} = \mathcal{T} = 2^{12}$. The results reported in Table VII indicate that our method outperforms both M-CALIC and CCSDS-123 in terms of lossless coding ($\Delta = 0$) for all sensors. In the near-lossless regime ($\Delta > 0$), the proposed approach outperforms M-CALIC when the USDQ is used and in most cases for the UQ. In particular, M-CALIC obtains slightly better results than our proposal only for images acquired with sensors Airs and Hyperion when UQ is used.

On the other hand, the proposed system always outperforms the near lossless extension of CCSDS-123 for both quantizers. In addition, in general for the same Δ value, the coding performance is better for the USDQ than for UQ. Although achieved bit-rates vary widely from image to image, low bit-rates can be obtained for all images with a modest value of PAEs (maximum absolute pixel error).

E. Visual Comparison

To evaluate visual performance, we show a region cropped from an image encoded at the “same” bit-rate by the proposed approach with the UQ, M-CALIC and CCSDS-123. For CCSDS-123, we employ the block-adaptive coder since we want to compare the images at a bit-rate lower than 1 bps. We note that none of the schemes compared here includes precise rate control. For this reason we have employed the following methodology: 1) encode an image using a variety of different quantization step sizes for each compression scheme, and 2) choose those encoded images that yield bit-rates as close as possible for the three algorithms. We note that a close match was not obtained in the case of CCSDS-123, so a step size was chosen to afford a higher bit-rate than that of the proposed approach, thus giving an advantage to CCSDS-123 in terms of visual performance.

The results of this process are shown in Fig. 5 for a crop from component 122 of the image “Avisir Calibrated Yellowstone sc00”. The bit rates obtained are 0.43, 0.42 and 0.50 for the

TABLE VII: Lossless ($\Delta = 0$) and near lossless ($\Delta > 0$) compression results for the proposed approach. For comparison, results for JPEG-LS, M-CALIC and CCSDS-123 are included. Both a UQ and a USDQ have been used in our proposed approach and in our near-lossless extension to CCSDS-123 to produce results for $\Delta > 0$. Results are reported in bits per sample (lower is better).

		CCSDS-123				JPEG-LS	M-CALIC	Our Proposal	
		with UQ		with USDQ				with UQ	with USDQ
		Sample adaptive	Block adaptive	Sample adaptive	Block adaptive				
Sensor	Δ values								
AC	$\Delta = 0$	3.73	3.91	3.73	3.91	6.41	3.87	3.71	3.71
	$\Delta = 10$	1.20	0.97	1.20	0.94	2.45	0.76	0.62	0.60
	$\Delta = 20$	1.10	0.74	1.10	0.72	1.81	0.50	0.38	0.36
	$\Delta = 30$	1.07	0.64	1.07	0.63	1.48	0.40	0.28	0.27
AU	$\Delta = 0$	5.06	5.23	5.06	5.23	7.47	5.13	5.03	5.03
	$\Delta = 10$	1.69	1.68	1.85	1.78	3.41	1.46	1.39	1.52
	$\Delta = 20$	1.35	1.19	1.40	1.21	2.68	0.95	0.87	0.90
	$\Delta = 30$	1.24	0.98	1.26	0.99	2.30	0.73	0.65	0.67
A	$\Delta = 0$	4.29	4.48	4.29	4.48	6.85	4.28	4.27	4.27
	$\Delta = 10$	1.23	1.12	1.18	0.96	2.62	0.73	0.76	0.62
	$\Delta = 20$	1.10	0.75	1.07	0.65	1.86	0.41	0.39	0.30
	$\Delta = 30$	1.06	0.63	1.05	0.56	1.50	0.31	0.28	0.22
C	$\Delta = 0$	4.97	5.15	4.97	5.15	6.79	4.91	4.87	4.87
	$\Delta = 10$	1.47	1.48	1.51	1.47	2.64	1.12	1.10	1.10
	$\Delta = 20$	1.25	1.06	1.25	1.03	1.94	0.68	0.67	0.64
	$\Delta = 30$	1.17	0.88	1.18	0.87	1.58	0.52	0.49	0.48
Cr	$\Delta = 0$	4.40	4.50	4.40	4.50	5.10	6.91	4.18	4.18
	$\Delta = 10$	1.64	1.66	1.63	1.42	1.83	2.75	1.26	0.99
	$\Delta = 20$	1.43	1.34	1.39	1.05	1.47	2.02	0.92	0.64
	$\Delta = 30$	1.34	1.17	1.30	0.89	1.29	1.64	0.76	0.50
H	$\Delta = 0$	4.37	4.57	4.37	4.57	6.24	4.80	4.28	4.28
	$\Delta = 10$	1.38	1.45	1.21	1.08	2.74	1.02	1.06	0.68
	$\Delta = 20$	1.24	1.19	1.09	0.74	2.06	0.52	0.77	0.35
	$\Delta = 30$	1.18	1.04	1.06	0.63	1.68	0.33	0.62	0.25
M	$\Delta = 0$	2.81	2.97	2.81	2.97	4.24	5.18	2.69	2.69
	$\Delta = 10$	1.26	1.21	1.11	0.74	1.38	1.32	0.76	0.33
	$\Delta = 20$	1.17	1.01	1.07	0.60	1.20	0.78	0.56	0.20
	$\Delta = 30$	1.14	0.89	1.06	0.55	1.00	0.53	0.46	0.15
Average	$\Delta = 0$	4.23	4.40	4.23	4.40	6.16	5.01	4.15	4.15
	$\Delta = 10$	1.41	1.37	1.39	1.20	2.44	1.31	1.00	0.83
	$\Delta = 20$	1.23	1.04	1.20	0.86	1.86	0.84	0.65	0.48
	$\Delta = 30$	1.17	0.89	1.14	0.73	1.55	0.64	0.51	0.36

proposed approach, M-CALIC and CCSDS-123, respectively. The reader is invited to zoom in to see the specific visual artifacts arising from the different compression schemes. The figure indicates that the image obtained by the proposed approach has higher visual quality than those of (near lossless) CCSDS-123 and M-CALIC. In particular, the proposed approach preserves edges and textures very well, while M-CALIC results in smoothness and loss of texture. CCSDS-123 also removes texture, but also introduces an annoying “banding” effect, due to the high step size required to reach 0.50 bps.

VI. CONCLUSIONS

This manuscript proposes an entropy encoder based on an efficient definition for a context model and associated strategy

to estimate probabilities for use in fixed length arithmetic encoder using low-cost bitwise operations. These contributions are incorporated in a coding approach that employs the predictor included in CCSDS-123. A near-lossless quantizer has also been deployed. The entropy encoder works on a line-by-line and bitplane-by-bitplane scanning order. Experimental results indicate that the use of a single neighbor for the context formation is enough to properly exploit the contextual information in the arithmetic encoder, and that it is possible to estimate the probability employing bitwise operations without penalizing the coding efficiency. Further results indicate that, on average, our proposal improves the current standard version of CCSDS-123 for lossless coding by more than 0.1 bps. Compared with M-CALIC, our proposal provides an average improvement of 0.86 bps for lossless, whereas for near-lossless

the benefit ranges from 0.13 to 0.31 bps, depending on the allowed peak absolute error.

REFERENCES

- [1] Consultative Committee for Space Data Systems (CCSDS). [Online]. Available: <http://www.ccsds.org>
- [2] *Lossless Data Compression*, CCSDS-121.0-B-1 Blue Book, May 1997.
- [3] *Image Data Compression*, CCSDS-122.0-B-1 Blue Book, Nov 2005.
- [4] *Lossless Multispectral & Hyperspectral Image Compression*, CCSDS-123.0-B-1 Blue Book, May 2012.
- [5] ISO/IEC, "JPEG-LS lossless and near-lossless compression for continuous-tone still images," 1999.
- [6] E. Magli, G. Olmo, and E. Quacchio, "Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 1, pp. 21–25, Jan. 2004.
- [7] F. Rizzo, C. Bruno, G. Motta, and J. A. Storer, "Low-complexity lossless compression of hyperspectral imagery via linear prediction," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 138–141, 2005.
- [8] G. Carvajal, B. Penna, and E. Magli, "Unified lossy and near-lossless hyperspectral image compression based on jpeg 2000," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, pp. 593–597, Oct 2008.
- [9] C. W. Chen, T. C. Lin, S. H. Chen, and T. K. Truong, "A near lossless wavelet-based compression scheme for satellite images," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 6, March 2009, pp. 528–532.
- [10] S.-C. Tai, T.-M. Kuo, C.-H. Ho, and T.-W. Liao, "A near-lossless compression method based on CCSDS for satellite images," *International Symposium on Computer, Consumer and Control*, 2012.
- [11] J. Song, Z. Zhang, and X. Chen, "Lossless compression of hyperspectral imagery via rls filter," *IEEE Electronics Letters*, vol. 49, no. 16, pp. 992–994, Aug 2013.
- [12] 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, Sept 2014.
- [13] D. Valsesia and E. Magli, "A novel rate control algorithm for onboard predictive coding of multispectral and hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6341–6355, 2014.
- [14] J. Beerten, I. Blanes, and J. Serra-Sagrista, "A fully embedded two-stage coder for hyperspectral near-lossless compression," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp. 1775–1779, Aug 2015.
- [15] J. Wu, K. Wanqiu, M. Jarno, and H. Bormin, "Lossless compression of hyperspectral imagery via clustered differential pulse code modulation with removal of local spectral outliers," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2194–2198, 2015.
- [16] F. Gao and G. Shuxu, "Lossless compression of hyperspectral images using conventional recursive least-squares predictor with adaptive prediction bands," *SPIE Journal of Applied Remote Sensing*, vol. 10, no. 1, pp. 015 010–015 010, 2016.
- [17] N. Amrani, J. Serra-Sagrista, V. Laparra, M. W. Marcellin, and J. Malo, "Regression wavelet analysis for lossless coding of remote-sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5616–5627, Sept 2016.
- [18] A. B. Kiely and M. A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2672–2678, Aug 2009.
- [19] X. Wu and P. Bao, " L_∞ constrained high-fidelity image compression via adaptive context modeling," *IEEE Trans. Image Process.*, vol. 9, pp. 536–542, April 2000.
- [20] B. Aiazzi, L. Alparone, and S. Baronti, "Context modeling for near-lossless image coding," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 77–80, 2002.
- [21] M. Slattery and J. Mitchell, "The Qx-coder," *IBM Journal of Research and Development*, vol. 42, no. 6, pp. 767–784, Nov. 1998.
- [22] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, July 2003.
- [23] *Information technology - Lossy/lossless coding of bi-level images*, ISO/IEC Std. 14492:2001, Dec. 2001.
- [24] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15444-1, Dec. 2000.
- [25] *H.264 - Advanced video coding for generic audiovisual services*, ITU-T Std., May 2003.
- [26] *H.265 - High efficiency video coding*, ITU-T Std., Apr. 2015.
- [27] M. D. Reavy and C. G. Boncelet, "An algorithm for compression of bilevel images," *IEEE Trans. Image Process.*, vol. 10, no. 5, p. 669676, may 2001.
- [28] F. Auli-Llinas, "Context-adaptive binary arithmetic coding with fixed-length codewords," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1385–1390, Aug. 2015.
- [29] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards and practice*. Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 2002.
- [30] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
- [31] A. Kiely and M. Klimesh, "A new entropy coding technique for data compression," Jet Propulsion Laboratory, California Institute of Technology, Tech. Rep., Jun. 2001.
- [32] *Lossless Multispectral & Hyperspectral Image Compression*, CCSDS-120.2-G-1 Green Book, Dec 2015.
- [33] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Prentice-Hall, 1984.
- [34] X. Wu and N. Memon, "Context-based lossless interband compression-extending calic," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 994–1001, Jun 2000.
- [35] E. Auge, J. E. Sanchez, A. Kiely, I. Blanes, and J. Serra-Sagrista, "Performance impact of parameter tuning on the ccsds-123 lossless multi- and hyperspectral image compression standard," *Journal of Applied Remote Sensing*, vol. 7, no. 1, pp. 074 594–074 594, 2013.