Constant-SNR, rate control and entropy coding for predictive lossy hyperspectral image compression

Constant SNR, Rate Control, and Entropy Coding for Predictive Lossy Hyperspectral Image Compression

Marco Conoscenti, Riccardo Coppola, and Enrico Magli, *Senior Member, IEEE*

***Abstract*—Predictive lossy compression has been shown to represent a very flexible framework for lossless and lossy onboard compression of multispectral and hyperspectral images with quality and rate control. In this paper, we improve predictive lossy compression in several ways, using a standard issued by the Consultative Committee on Space Data Systems, namely CCSDS-123, as an example of application. First, exploiting the flexibility in the error control process, we propose a constant-signal-to-noise-ratio algorithm that bounds the maximum relative error between each pixel of the reconstructed image and the corresponding pixel of the original image. This is very useful to avoid low-energy areas of the image being affected by large errors. Second, we propose a new rate control algorithm that has very low complexity and provides performance equal to or better than existing work. Third, we investigate several entropy coding schemes that can speed up the hardware implementation of the algorithm and, at the same time, improve coding efficiency. These advances make predictive lossy compression an extremely appealing framework for onboard systems due to its simplicity, flexibility, and coding efficiency.**

***Index Terms*—Hyperspectral image coding, lossy compression predictive coding, multispectral image compression, rate control.**

I. Introduction

COMPRESSION of multispectral and hyperspectral images has gained increasing interest as new sensors are acquiring large amounts of spatial and spectral information. Since the communication capacity of downlink channels between remote platforms and ground stations is not increasing accordingly, compression plays a crucial role in reducing the amount of data to be transmitted and maximizing the scientific return of a given remote sensing mission.

Compression can be of different types, namely lossless and lossy. Lossless compression is often based on a mathematical model to predict pixel values and encode only their prediction residuals. Adaptive linear prediction is often used [2]–[9]. An effective technique relies on the least mean squares filter [10] with the sign algorithm [11] for weight update; this has been employed in [12] for hyperspectral image compression, as well as in the recent standard issued by the Consultative Committee on Space Data Systems (CCSCS), namely CCSDS-123 [13]. Other methods have also been devised, based on edge detection [14], clustering [15], [16], or vector quantization [17]. Recently, the distributed source coding paradigm has also been used to achieve low-complexity lossless and near-lossless compression [18], [19].

Lossy compression has traditionally been performed following the transform coding paradigm, whereby a linear transform of the data is used to achieve energy compaction and hence transmit few carefully chosen transform coefficients, see, e.g., [20]–[29]. Predictive compression also lends itself well to lossy compression, where the prediction residuals are fed into a quantization feedback loop before entropy coding. The quantization step size determines the amount of information losses with respect to the original image. Choosing a different quantization step size for each pixel enables more general and flexible quality policies that may suit specific mission requirements, whereas transform coding typically controls only the average error over the whole image. Near-lossless compression can also be obtained using transform coding, although this requires onboard decoding [30], [31]. Transform coding allows to accurately control the rate in a simple manner due to the simple relation between rate and quantized transform coefficients [20], [32]. Rate control is harder to achieve using predictive compression because of the intricate mathematical relations between the rate and the quantized prediction residuals.

In [1], it is shown that predictive compression enables both rate and quality control in an efficient way. In [1] and [33], this paradigm has been applied to the CCSDS-123 recommendation, taking a first step toward a very flexible compression algorithm that can accommodate a wide range of requirements, from lossless up to several different modes of lossy compression. However, the framework in [1] still has a few open problems. This paper advances the state of the art by providing innovative quality control, rate control, and entropy coding techniques that significantly upgrade [1], yielding a simple and fully fledged solution for onboard compression of multispectral and hyperspectral images. It is worth noting that, while these concepts are applied to the CCSDS-123 framework, the basic ideas are very general and can be applied to any predictive scheme with little or no modifications.

Quality control: While near-lossless compression bounds the maximum absolute reconstruction error on any pixel of the image, in many cases, a much sought after feature is the ability of bounding the *relative* error. This is because a fixed maximum absolute error will yield a signal-to-noise ratio (SNR)

Manuscript received March 21, 2016; revised July 5, 2016; accepted July 29, 2016. This work was supported in part by the Italian Space Agency through the "Solar Orbiter: Supporto scientifico alla realizzazione di METIS e SWA" and in part by the European Space Agency–European Space Research and Technology Center (ESA–ESTEC) under Grant 107104 "HYDRA."

M. Conoscenti is with Nexa Center for Internet and Society, Department of Control and Computer Engineering, Politecnico di Torino, 10129 Torino, Italy (e-mail: marco.conoscenti@polito.it).

R. Coppola and E. Magli are with Politecnico di Torino, 10129 Torino, Italy.

Digital Object Identifier 10.1109/TGRS.2016.2603998

that varies over the image and is small in low-energy areas. Conversely, bounding the relative error leads to a constant-SNR approach, which is more desirable in many applications, see, e.g., [34]. However, bounding the relative error is more difficult because the relative error depends on the value of the original pixel, which is not known at the decoder in case of lossy compression. In this paper, we introduce an algorithm for choosing quantization step sizes in a predictive lossy compression scheme, which can achieve approximate or exact bounding of the relative error, leading to effective constant-SNR compression.

Rate control: The rate control algorithm in [1] solves a multivariate optimization problem to find the optimal quantization step sizes for a set of spatial/spectral image blocks minimizing reconstruction distortion for a given target rate. While the number of operations needed by the algorithm is not very large, the iterative nature of the optimization process makes it difficult to implement in hardware. In this paper, we propose a new simplified rate control algorithm that employs univariate optimization. The resulting algorithm does not require iterative optimization, thereby providing lower complexity and also improved coding efficiency with respect to [1].

Entropy coding: Range encoding [35] has been shown in [1] to provide improved performance with respect to Golomb coding, even when splitting the prediction residuals into subalphabets as in [36]. Moreover, in [37], a field-programmable gate array implementation of a range encoder on space-qualified hardware has been reported, whose throughput exceeds 40 MB/s and is hence to be considered suitable for onboard compression. In this paper, we investigate different setups for the entropy coding stage. First, we consider a range encoder that is applied to bit planes of the prediction residuals; a bit-plane encoder neglects correlation between adjacent bit planes but is fast at learning the probability model for each bit plane. We show that the resulting encoder is simpler than and outperforms the range encoder in [1]. Second, we investigate a simple scheme where a Golomb code is initially used, and a single binary range encoder is employed to further compress the concatenated Golomb codewords. The motivation of this scheme is as follows. As is known, the Golomb code does not work well at bit rates below 2 bits per pixel per band (bpppb) since its minimum codeword length is 1 bit. At those bit rates, therefore, the distribution of 0's and 1's in the output stream is expected to be different from 0.5; therefore, a binary range encoder can take advantage of this and partially compensate for the suboptimality of the first encoder. This is not the only possible solution as a more conventional approach employs a run-length mode to achieve the lower bit rates, e.g., in [38]. However, a run-length mode is undesirable in hardware as it requires stopping the Golomb encoder and entering/exiting a specific run mode. On the other hand, in a hardware implementation, a Golomb and range encoder can be effectively pipelined, thereby streamlining the algorithm operation flow and achieving a higher throughput than a fully sequential solution.

This paper is organized as follows. In Section II, we review some background material, particularly the CCSDS-123 standard and its extension to lossy compression with rate control. In Section III, we describe the proposed constant-SNR compression scheme. In Section IV, we propose the new simplified rate control algorithm. In Section V, we discuss the entropy coding approaches based on bit-plane range encoder and cascaded Golomb–range encoder. Finally, in Section VI, we draw some conclusions.

II. Review

In the following, we briefly review the CCSDS-123 lossless compression recommendation, its extension to lossy compression, and the rate control algorithm proposed in [1]. More details can be found in the tutorial paper [39] as well as in [13].

A. CCSDS-123 Standard

As is the case of many lossless compression schemes, CCSDS-123 is based on a predictor followed by an encoder of the prediction residuals. Given the original pixel $x_{i,j,k}$ (row $i \in [1, N_{\mathrm{rows}}]$, column $j \in [1, N_{\mathrm{columns}}]$, band $k \in [1, N_{\mathrm{bands}}]$), a predicted pixel value $\widetilde{x}_{i,j,k}$ is first computed as a function of neighboring pixel values. In the following, we will also use the notation $l = (i, j, k)$ and denote a pixel value as x_l to simplify the notation when appropriate.

The predicted pixel $\widetilde{x}_{i,j,k}$ is computed based on neighboring pixels, both in a spatial or in a spectral sense. These neighboring pixels are combined to produce a local sum $\sigma_{i,j,k}$. Normally, four spatial neighbors are used to compute the local sum, i.e., $\sigma_{i,j,k} = x_{i-1,j-1,k} + x_{i-1,j,k} + x_{i-1,j+1,k} + x_{i,j-1,k}$. This local sum is then scaled and used to predict $x_{i,j,k}$. In particular, the difference between the local sum and the corresponding scaled original pixels is tracked and stored in a local difference vector $\mathbf{U}_{i,j,k}$ for some samples. In the *full* prediction mode $\mathbf{U}_{i,j,k}$ is expressed as

$$\mathbf{U}_{i,j,k} = \begin{bmatrix} 4 \cdot x_{i-1,j,k} - \sigma_{i,j,k} \\ 4 \cdot x_{i,j-1,k} - \sigma_{i,j,k} \\ 4 \cdot x_{i-1,j-1,k} - \sigma_{i,j,k} \\ 4 \cdot x_{i,j,k-1} - \sigma_{i,j,k-1} \\ 4 \cdot x_{i,j,k-2} - \sigma_{i,j,k-2} \\ \vdots \\ 4 \cdot x_{i,j,k-P} - \sigma_{i,j,k-P} \end{bmatrix}$$

where the first three components represent spatial directional local differences, and the other components represent spectral local differences. An inner product is then taken between the local difference vector $\mathbf{U}_{i,j,k}$ and a weight vector $\mathbf{W}_{i,j,k}$, yielding an estimation of the local differences $\hat{S}_{i,j,k} = \mathbf{W}_{i,j,k}^T \mathbf{U}_{i,j,k}$. This estimation is used to generate the predicted value and hence the prediction residual. The sign algorithm updates the weight vector to be used for the next pixel, attempting to adaptively learn the image statistics to minimize the energy of the prediction residual. Other prediction modes are also available (see [13]). A prediction residual is computed and then mapped to a nonnegative integer that is passed on to the entropy coding stage. This stage can be carried out on a sample-adaptive or block-adaptive basis. The sample-adaptive Golomb coding stage employs a Golomb code, whose parameter is computed adaptively; this typically provides better coding efficiency than the legacy block-adaptive stage.

B. Extension to Lossy Compression and Rate Control

Extending the algorithm above to near-lossless compression is very simple as the addition of a quantization feedback loop allows obtaining an algorithm that is able to provide bounded absolute maximum error between any original pixel and the reconstructed one. In particular, we use the following notation: $\widetilde{x}_l$ is the predicted pixel value; $e_l = x_l - \widetilde{x}_l$ is the prediction residual; q_l is the quantization step size applied to that residual; m_l is the mapped quantized prediction residual; and x_l^R is the reconstructed pixel. When it is appropriate and it does not cause confusion, we also use the notation q_i to denote a quantization step size to be employed to all pixels of data unit i, which can be a block (as is the case later in this section) or a slice (see Section IV).

Near-lossless compression, however, is unable to provide rate control. In [1], a solution to this problem has been proposed, which performs greedy allocation of quantizers to achieve the desired target rate R_T. Rate control works on a slice-by-slice basis, where a "slice" is defined as a predefined number of lines (e.g., 16) with all their spectral channels. Each slice is divided into nonoverlapping 16×16 blocks (except for blocks at image boundaries, which are allowed to be smaller). An individual quantization step size is computed for each block in each spectral channel, to achieve a rate as close as possible to R_T. The rate control algorithm performs the following steps.

1) *Training stage*: a rate–distortion model is computed for each block i in each spectral channel of the slice as a function of the quantization step size q_i to be used for the block and of parameter $\Lambda_i = \sqrt{2/\sigma_i^2}$ depending on the variance σ_i^2 of the unquantized prediction residuals of the block. The latter is estimated by running the lossless predictor on a small number of lines in the slice. This process defines rate–distortion functions $R(q_i, \Lambda_i)$ and $D(q_i, \Lambda_i)$ which, given a quantization step size q_i chosen for block i and the variance σ_i^2, provide an estimate of the rate needed to code the block and the incurred distortion. The rate–distortion function of a Laplacian source is employed since the Laplacian distribution is a good model for prediction residuals, and the corresponding rate–distortion function is known in closed form. Then, the final quantization step sizes q_i are obtained as follows.
2) *Optimization stage—step 1*: An initial set of quantization step sizes q_i are calculated, which achieve the target rate but are suboptimal in terms of distortion. This can be done considering first the lossless compression case, i.e., $q_i = 1$ for $i = 1, \ldots, N$, being N the number of blocks belonging to a slice and the corresponding rates $R(1, \Lambda_i)$ for each block. This rate–distortion point is then "projected" onto the set of points that achieve the desired target rate, namely $\sum_{i=1}^{N} R(q_i, \Lambda_i) = R_T$. The unknown terms q_i can be calculated using standard optimization techniques. The obtained q_i's achieve the target rate but are not necessarily very good in terms of minimizing distortion.
3) *Optimization stage—step 2*: When the target rate has been achieved, the algorithm makes changes to the q_i's to solve the constrained problem, i.e.,

$$\min \sum_{i=1}^{N} D(q_i, \Lambda_i) \text{ subject to } \sum_{i=1}^{N} R(q_i, \Lambda_i) \leq R_T.$$

In particular, a greedy algorithm makes local adjustments to q_i's aimed at promoting low-distortion allocations of the quantization step sizes, employing the rate–distortion models of all blocks in the slice. This procedure employs an iterative perturbation of quantization step sizes, which is terminated when no further improvement is achieved. More details can be found in [1].

Finally, the algorithm employs feedback from one slice to the next, using information on the actual rate produced encoding a slice to update the target rate for future slices, thereby compensating for inaccuracies of the rate control process.

TABLE I
TEST IMAGES

Name	N_{rows}	$N_{columns}$	N_{bands}
AGRICULTURE	1024	1024	6
AIRS-GRAN9	135	90	1501
AVIRIS-SC0	512	680	224
CASI-T0477F06-NUC	1225	406	72
CRISM-SC214-NUC	510	640	545
EO-1-GEO_SAMPLE-NUC	1024	256	242
FRT00009326-07-VNIR	512	640	107
MODIS-MOD01-250M	8120	5416	2
MONTPELLIER	224	2456	4
T0477F06-RAD-RSS	1225	406	72

C. Test Images

The test images used in this paper are a subset of the data employed by CCSDS for the performance assessment of lossy compression algorithms. They encompass multispectral, hyperspectral, and ultraspectral images, showing that the proposed techniques are able to operate in a variety of different conditions. In particular, we consider ultraspectral images captured by Infrared Atmospheric Sounding Interferometer and AIRS sensors, hyperspectral images captured by Compact Airborne Spectrographic Imager, AVIRIS and Hyperion sensors, and multispectral images captured by MODIS, Landsat, Meteosat Second Generation, and Pleiades sensors. The images and their sizes are listed in Table I.

III. QUALITY CONTROL

This section describes the proposed constant-SNR coding scheme and shows the advantages of such a technique.

Examining the behavior of a near-lossless encoder, it can be seen that, in the case of small sample values, a simple upper bound to the absolute reconstruction error may be unsatisfactory as it may result in the reconstruction error being very large with respect to the sample itself, or even bigger.

The new approach proposed in the following computes an adaptive quantization step size that depends on the value of each sample, in order to upper-bound the reconstruction error to a user-defined maximum percentage of the sample value itself. Since the original sample is not available at the decoder side, the predicted value is utilized for the computations. A parameter

$W \geq 0$ is passed as user input to the algorithm, representing the maximum relative reconstruction error accepted. The quantization step size for pixel l with coordinates i, j, k is thus obtained using the following:

$$q_l = 2 * \lfloor W * |\widetilde{x}_l| \rfloor + 1.$$

A. Enhancements to the Basic Algorithm

The simple aforementioned operations are not sufficient to guarantee a bounded relative reconstruction error for any pixel l, defined as $E_l = |x_l^R - x_l|/|x_l|$, since the use of the predictions introduces some suboptimalities. Essentially, because the predicted value is close but not generally identical to the original sample value, a certain number of samples will be affected by a relative reconstruction error bigger than the acceptable threshold. To address this problem, some improvements have been added to the basic approach.

- We take some "safety margin," i.e., we set the algorithm to provide a smaller relative error than is actually requested by the user. In practice, this is obtained by multiplying W by another constant, $0 \leq P \leq 1$. This has the effect of shrinking the quantization step size for all the pixels, reducing significantly the amount of samples exceeding the reconstruction error threshold. Since taking a safety margin indeed increases the bit rate, one has to be careful to not overdo it; in practice, $P = 0.9$ has been found to be a good compromise.
- Lossless coding of the first row of each band of the image has proved to be a good way to reduce considerably the number of outliers, at the price of an acceptable increase in the size of the compressed image. This is because, on the first row, the predictor is less precise due to the lack of samples on the row above, leading to larger prediction errors. Performing lossless compression of these samples nullifies this effect.
- Finally, we have found that a high increase in the reconstruction error occurs when the predictor is significantly bigger than the corresponding original sample. To reduce the effect of erroneous estimates, it is worth using a lossless coding for those samples whose current predictor is bigger than twice the value of the previous reconstructed sample. Note that, since the value of the reconstructed sample depends on this condition, the condition is evaluated using the previous reconstructed sample, so that the decoder can perform the same calculations.

Note that all these modifications to the algorithm do not require any signaling since the decoder can apply the same algorithm to calculate the correct quantization step used at the encoder.

B. Reconstruction Error Compensation

To cope with the unavoidable presence of reconstruction errors bigger than the imposed threshold, an explicit correction step can be inserted in the algorithm in case the user wants to enforce a strict constraint that no sample at all must have a relative error above the threshold. To this end, for each pixel whose reconstruction error still exceeds the threshold, the encoder can calculate the smallest offset that has to be added to the reconstructed sample to bring the reconstruction error below the threshold.

For each sample to be adjusted, two values have to be communicated to the decoder: its position inside the image in the form of an unsigned integer, and the correction offset in the form of a short integer. These six-byte data records are stored by the encoder in an auxiliary repair file, which is then read by the decoder at the end of its operations, and used to adjust the reconstructed samples.

The repair file itself can be subject to lossless compression as it indeed shows some redundancy. In particular, it can be seen that the offsets follow a Laplacian distribution, deprived of the zero value: a binary bit-plane coder can therefore be applied to the offset values after subtracting one from the positive values. To encode the indexes, we apply a binary range coder to the difference between the current index and the previous one.

With a coefficient $P = 0.9$ and a reasonable reconstruction error threshold (above 1%), we have found that the number of samples exceeding the threshold is relatively small, so that the repair file has a small incidence on the overall bit rate generated by the encoder.

The full set of operations done at encoder side is shown in Algorithm 1.

Algorithm 1 Quality control algorithm for sample l (coordinates i, j, k)

if $i = 1$ or $\widetilde{x}_l \geq 2|x_{l-1}^R|$ **then**
    Set $q_l = 1$
else
    Set $q_l = 2 * \lfloor W * |\widetilde{x}_l| \rfloor + 1$
end if
Compute mapped quantized residual m_l using q_l
Encode m_l
Set $E_l = |x_l^R - x_l|/|x_l|$
if $E_l > W$ **then**
    if $x_l^R > x_l$ **then**
        Set $O_l = \lfloor x_l * (1 + W) \rfloor - x_l^R$
    else
        Set $O_l = \lceil x_l * (1 - W) \rceil - x_l^R$
    end if
    Append l and O_l to repair file
end if

C. Error Analysis

In the following, we show that the proposed algorithm can indeed bound the relative error between original and reconstructed pixels, i.e., $E_l \leq W$; for this analysis, we set $P = 1$, i.e., we do not employ any safety margin. This is equivalent to verifying that $|x_l^R - x_l| \leq W|x_l|$. We note that, in a differential pulse-code modulation scheme, the reconstruction error on x_l is exactly the same as the reconstruction error on the prediction residual, which is quantized with step size q_l, and that if q_l is an odd number, then $q_l = 2\text{MAD}_l + 1$, where MAD_l is the maximum absolute difference (MAD) between the original

TABLE II
QUALITY CONTROL ALGORITHM—RESULTS

Image	Error Threshold	SNR	Bitrate	Repair file bitrate	Relative no. of samples above threshold	Average relative error
AGRICULTURE	0.005	96.36	3.81	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$
	0.01	47.37	3.57	0.094	0.005	0.001
	0.05	31.46	1.38	0.219	0.013	0.022
AIRS-GRAN9	0.005	52.19	0.80	$< 10^{-3}$	$< 10^{-3}$	0.002
	0.01	47.14	0.40	$< 10^{-3}$	$< 10^{-3}$	0.004
	0.05	35.05	0.09	0.043	0.002	0.015
AVIRIS-SC0	0.005	51.14	1.65	0.007	$< 10^{-3}$	0.002
	0.01	45.59	1.07	0.010	$< 10^{-3}$	0.004
	0.05	32.80	0.40	0.123	0.003	0.018
CASI-T0477F06-NUC	0.005	51.07	3.01	0.021	0.001	0.088
	0.01	45.13	2.02	0.025	0.001	0.095
	0.05	32.38	0.70	0.169	0.009	0.039
CRISM-SC214-NUC	0.005	51.18	1.51	0.002	$< 10^{-3}$	0.002
	0.01	45.79	0.88	0.002	$< 10^{-3}$	0.004
	0.05	34.02	0.15	0.046	0.002	0.016
EO-1-GEO_SAMPLE-NUC	0.005	51.02	1.86	0.001	$< 10^{-3}$	0.002
	0.01	45.26	1.11	0.001	$< 10^{-3}$	0.005
	0.05	34.38	0.10	0.041	0.002	0.015
FRT00009326-07-VNIR	0.005	51.01	2.49	0.019	$< 10^{-3}$	0.002
	0.01	44.96	1.67	0.018	$< 10^{-3}$	0.005
	0.05	32.85	0.40	0.138	0.008	0.020
MODIS-MOD01-250M	0.005	51.00	4.08	0.118	0.007	0.002
	0.01	44.97	3.21	0.149	0.009	0.005
	0.05	31.59	1.38	0.290	0.016	0.022
MONTPELLIER	0.005	51.00	4.40	0.170	0.010	0.002
	0.01	44.97	3.44	0.186	0.011	0.005
	0.05	31.18	1.55	0.338	0.017	0.022
T0477F06-RAD-RSS	0.005	51.03	3.65	0.083	0.005	0.002
	0.01	45.11	2.76	0.094	0.005	0.005
	0.05	32.58	1.17	0.251	0.013	0.025

and reconstructed pixel l. Then, one can easily show that $|x_l^R - x_l| \leq (q_l - 1)/2 = \lfloor W * |\widetilde{x}_l| \rfloor \leq W|\widetilde{x}_l|$. If $|\widetilde{x}_l| \leq |x_l|$, then $|x_l^R - x_l| \leq W|x_l|$ and the error satisfies the bound. If this does not happen, the reconstruction error compensation stage kicks in. For example, if $E_l > W$ and $x_l^R > x_l$, then the reconstructed value after compensation is $x\prime_l^R = x_l^R + O_l = x_l^R + \lfloor x_l * (1+W) \rfloor - x_l^R = \lfloor x_l * (1+W) \rfloor$. Hence, $|x\prime_l^R - x_l| = |\lfloor x_l * (1+W) \rfloor - x_l| \leq |x_l * (1+W) - x_l| = W|x_l|$, which proves the bound on the relative error. An analogous calculation can be made if $E_l > W$ and $x_l^R < x_l$.

D. Results

The results collected over the CCSDS image test set[1] show that the constant-SNR approach satisfies the needs highlighted in the introduction and provides very good performance. The number of samples whose error is above the threshold is rather small for the majority of the images of the set, even without the repair file. It has been found that a significant reduction of the size of the repair file can be obtained after the application of a compression step, which makes the size essentially negligible. In Table II, the results are shown for a few images of the test set. Different values of the threshold have been used, ranging from $W = 0.005$ to $W = 0.5$. The table also shows the percentage of samples above the error threshold and the average relative error, before the application of the repair file.

[1]The test set can be found at http://cwe.ccsds.org/sls/docs/sls-dc/123.0-B-Info/TestData.

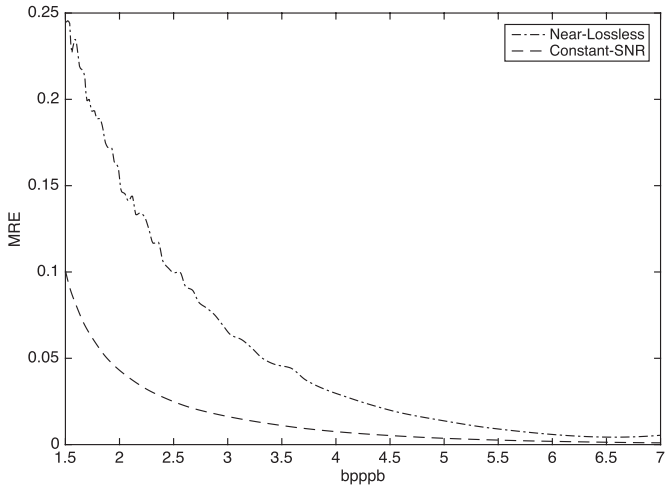

Fig. 1. Maximum relative error with respect to bit rate for MONTPELLIER.

A natural term of comparison for the results of the developed algorithm is a simple near-lossless extension of the CCSDS-123 encoder, obtained by using the same quantization step size for all pixels of the image in the feedback loop, which bounds the absolute instead of the relative error. The graph in Fig. 1, obtained by applying the two algorithms to the MONTPELLIER image, shows that, for the same bit rate, the maximum relative error is significantly smaller if the constant-SNR algorithm is used. Note that, throughout this paper, the SNR is defined as $10\log_{10}(\sum_{i,j,k} x_{i,j,k}^2 / \sum_{i,j,k} (x_{i,j,k} - x_{i,j,k}^R)^2)$.

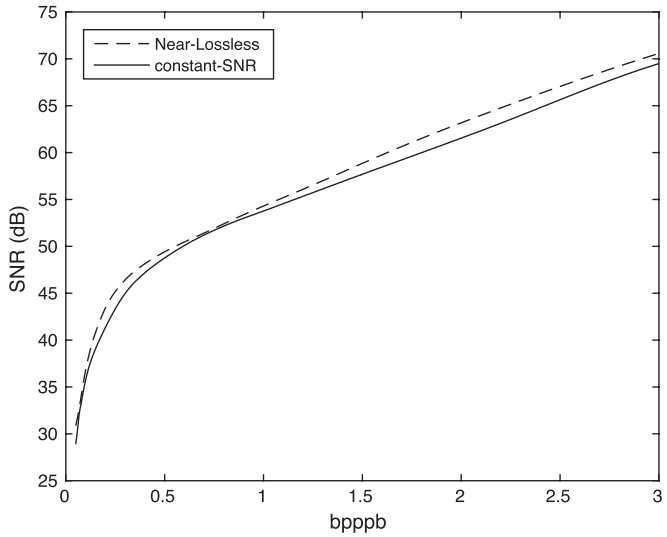

Fig. 2. Rate–distortion curves comparison for AIRS-GRAN9.

While imposing the new constraint of a limited relative reconstruction error, the constant-SNR algorithm shows a competitive rate–distortion curve. For some images of the test set and for certain bit rates, the new algorithm outperforms the near-lossless one; this is remarkable as near-lossless compression typically obtains very high SNR since SNR is based on the mean square error, which directly depends on the absolute and not the relative error. The comparison between the rate–distortion curve for near-lossless and quality control coder is shown in Fig. 2, relatively to AIRS-GRAN9 (including the bit rate of the repair file). As shown, the constant-SNR algorithm, even if it does not aim at maximizing the SNR for a given bit rate, achieves rate–distortion performance very close and occasionally better than that of the near-lossless algorithm.

IV. Rate Control

Here, we propose a new rate control algorithm that aims to simplify the algorithm in [1] in terms of computational complexity and memory usage.

A. Proposed Rate Control Algorithm

In the rate control algorithm described in [1], the image is subdivided into blocks, and a rate–distortion model is employed, described by functions $R(q_i, \Lambda_i)$ and $D(q_i, \Lambda_i)$ that return the rate and the distortion of block i, respectively, for a given value of the quantization step q_i applied to all pixels of the block and the parameter Λ_i depending on the variance of the unquantized prediction residuals of the block. For each spectral slice, the set of q_i's yielding the target rate R_T with minimum distortion are found as the solution to a multivariate optimization problem using an iterative method. A feedback-based mode adjusts the target rate for subsequent slices to compensate for inaccuracies of the rate control procedure; we call $r_t[i]$ the actual target rate that the feedback-based mode targets for slice i.

Unlike [1], in the rate control algorithm proposed here, the subdivision of the image into blocks has been removed, and the variance of prediction residuals is computed for each spatial line of the image in a given spectral band, using all the unquantized prediction residuals of the line. The key aspect of the proposed algorithm is that the same quantization step is applied to all the pixels of a spectral slice. The rationale of this design choice is that this leads to a remarkably lower computational cost than [1]. In fact, since the quantization steps are constant along the spectral slices, finding the optimal step for each slice does not require to solve a multivariate optimization problem nor to run the subsequent greedy optimization. The proposed algorithm simply inverts the rate–distortion function to find the desired quantization step q to be used for all pixels and all bands of the current spectral slice. For the same goal of reducing the computational complexity, the feedback-based rate update algorithm does not employ the mathematical model of [1] but simply tracks the number of bits used for the already coded slices.

The proposed rate control algorithm is detailed in Algorithm 2. It is executed for each slice $i \in [1, N_{\text{rows}}]$, and its goal is to find the quantization step size q_i, to be applied to all pixels of slice i, which produces the rate nearest to the target for that slice. As we process a given slice i, for each possible quantization step q, we exploit the rate–distortion model $R(q, \Lambda_{i,z})$ to compute the expected rate that would be obtained by applying q to all pixels of the spectral slice. In particular, $\Lambda_{i,z}$ corresponds to the variance of the unquantized residuals for all pixels in band z belonging to row i. We start with $q = 1$ (lossless coding) and check whether the corresponding rate r is lower than the target $r_t[i]$. If it is not, then we increase q until $r < r_t[i]$, and we pick the value of q, yielding the rate nearest to $r_t[i]$ (when this happens, Boolean variable $found$ becomes TRUE). This value, denoted q_i, is used to quantize all samples in the current spectral slice. The algorithm also employs variables r_{i-1} and q_{i-1}, which are, respectively, the rate used to encode the previous slice and the corresponding quantization step size.

Algorithm 2 Proposed rate control algorithm for spectral slice with row index i.

```
1: for q = 1 : Q_max do
2:     r = 0; found = FALSE
3:     for k = 1 : N_bands do
4:         r = r + R(q, Λ_{i,k})
5:     end for
6:     if r ≤ r_t[i] AND found = FALSE then
7:         if q = 1 then
8:             Break for at line 1
9:         else
10:            found = TRUE
11:        end if
12:    end if
13:    if found = TRUE then
14:        if |r − r_t[i]| > |r_{i−1} − r_t[i]| then
15:            q = q_{i−1}
16:            Break for at line 1
17:        end if
18:    end if
19:    q_i = q; r_i = r
20: end for
```

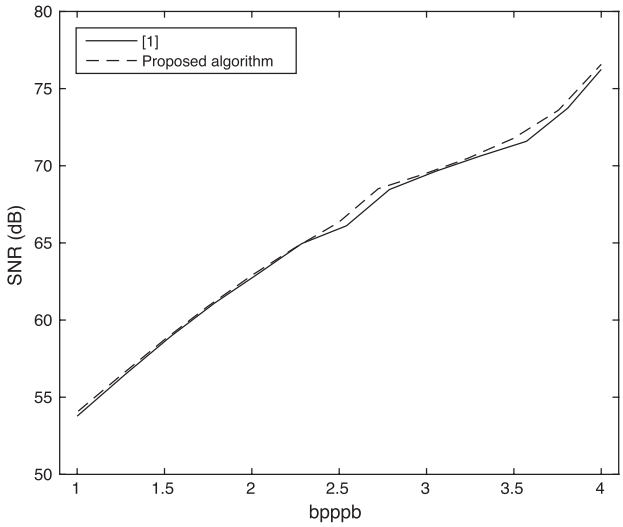

Fig. 3. Rate–distortion curves for AIRS-GRAN9 in mode B.

Now, we describe the feedback-based mode of the compressor. The main idea behind this algorithm is to adapt the target rate for the new slices according to the number of bits actually used for coding the previous slices. In particular, as shown in detail in Algorithm 3, r_a (which is initialized to zero) accumulates the differences between the original target rate R_T, given as input to the compressor, and the actual rate r_u of the already coded slices; therefore, its value will be negative if fewer bits than the target have been employed for the previous slices, and positive otherwise. The new target rate r_t to be used for S following slices is computed as the original target rate plus r_a; thus, it will be higher than the original target if the previous slices have been coded by fewer bits than the target, and lower otherwise.

Algorithm 3 Feedback-based mode algorithm to compute rate for slice $i+1$ after coding slice i.

1: update r_u
2: $r_a = r_a + (R_T - r_u)/S$
3: $r_t[i+1] = R_T + r_a$
4: **if** $r_t[i+1] < 0$ **then**
5: $\quad r_t[i+1] = R_T$
6: $\quad r_a = 0$
7: **end if**

B. Results

The results are reported in terms of SNR and MAD. The algorithm is assessed in two operating modes: mode A does not use rate feedback and is hence less accurate at achieving the target bit rate, whereas mode B usually achieves better results employing the feedback mode. Fig. 3 reports the rate–distortion curves for the image AIRS-GRAN9 in mode B. It can be noticed that, for this specific image, the algorithm provides better rate–distortion performance than [1], although it is much less computationally expensive. This aspect is more evident at higher bit rates.

In Table III, the results of the proposed algorithm are compared with [1] for some images of the test set. In mode A, in almost all cases, the proposed algorithm produced bit rates more similar to the target than [1]. For instance, this is very obvious for AGRICULTURE at 1 bpppb target, when [1] yields 0.797 bpppb, and the proposed algorithm 1.016 bpppb. In mode B, where the new feedback-based mode algorithm is activated, the bit rate produced is as close to the target as [1]. In terms of SNR, the proposed algorithm almost always reached higher values than [1]. To see this, one can look at CRISM-SC214-NUC at 4 bpppb target in mode A, where the proposed algorithm produced 78.52 dB of SNR, against 68.68 dB [1], where the improvement is also due to the ability to achieve a rate closer to the target (4.03 versus 3.86 bpp).

We note that for some images of the set, in mode A, the proposed algorithm has obtained bitrates below the target, especially at high bitrates. This imprecision is then overcome in mode B, when the feedback-based mode is activated and the final rate is very close to the target.

Finally, we provide software timing results for the new rate control algorithm in comparison with [1]. In particular, on an Intel Core i5-3230M CPU at 2.60 GHz, the compression algorithm in [1] with rate control in mode B takes 30.25 s to encode the AVIRIS-SC0 image, whereas the new rate control algorithm takes 28 s. Because of its much simpler structure, the new algorithm is expected to be a lot faster in hardware implementation.

C. Variance Estimation

As mentioned in Section IV-A, the variance of the prediction residuals for the pixels of a line is computed by using all the pixels of the line. This is computationally expensive since it requires double execution of the prediction part of the compression algorithm; indeed, a first execution is necessary to compute the unquantized prediction residuals employed for variance computation.

To cope with this, we show that it is possible to estimate the variance of a line by using only a fraction of pixels of the line. To show this, we have reported some results in Table IV, where N_e stands for the number of pixels used for variance estimation. One may notice that, even if we use just half of the pixels to estimate the variance—which in this case corresponds to every other pixel of the line—we obtain almost the same performance as in the case of using all the pixels.

V. Entropy Coding Methods

For onboard implementation, entropy coding has proven to be the most critical bottleneck in terms of achievable throughput. In this section, we propose some alternatives to the range coder used in [1], which provides very good compression performance but has larger computational and memory requirements with respect to the Golomb coder used in [13]. Two different techniques have been tried: The first one is a bit-plane entropy coder, implemented with a binary range coder for each of the bit planes composing the original set of mapped residuals; the second one is a cascaded combination of a Golomb coder (the same used in the original CCSDS-123 coder) and a simple binary range coder.

TABLE III
COMPARISONS BETWEEN PROPOSED ALGORITHM AND [1]

Image	Mode	R_T (bpppb)	ALGORITHM [1] Rate (bpppb)	ALGORITHM [1] SNR (dB)	ALGORITHM [1] MAD	PROPOSED ALGORITHM Rate (bpppb)	PROPOSED ALGORITHM SNR (dB)	PROPOSED ALGORITHM MAD
AGRICULTURE	A	1	0.780	26.34	13	1.02	28.80	38
	B	1	1.01	28.54	12	1.00	28.67	38
	A	2	1.85	35.89	4	1.78	35.40	7
	B	2	2.01	36.61	4	2.00	36.65	7
	A	4	3.78	55.23	1	3.82	62.50	2
	B	4	3.82	66.38	1	3.82	65.51	2
AIRS-GRAN9	A	1	1.06	54.48	10	1.16	55.47	9
	B	1	1.00	53.76	14	1.01	54.09	10
	A	2	2.08	63.52	4	1.96	62.57	4
	B	2	2.04	63.02	4	2.00	62.91	4
	A	4	4.10	78.25	1	4.16	81.01	1
	B	4	4.05	76.93	1	4.24	90.03	1
CRISM-SC214-NUC	A	1	0.80	44.65	15	0.94	46.40	15
	B	1	1.01	46.84	13	1	47.07	15
	A	2	1.90	54.37	4	1.67	52.93	6
	B	2	2.01	55.50	4	2.00	55.52	6
	A	4	3.86	68.68	1	4.03	78.52	1
	B	4	4.00	74.52	1	3.99	74.34	1
EO1-GEOSAMPLE-NUC	A	1	0.89	44.45	10	1.00	45.32	12
	B	1	1.01	45.18	11	1	45.25	12
	A	2	2.05	53.15	4	1.99	52.70	5
	B	2	2.01	52.67	4	2.00	52.81	5
	A	4	4.10	67.77	1	4.25	85.26	1
	B	4	4.01	65.63	1	4.01	66.11	1
MONTPELLIER	A	1	1.12	27.69	141	1.23	29.47	178
	B	1	1.03	26.69	166	1.01	27.43	178
	A	2	2.36	37.70	38	2.07	36.13	57
	B	2	2.09	35.45	71	2.00	35.62	57
	A	4	4.33	51.49	7	3.87	48.56	12
	B	4	4.09	49.79	10	4.00	49.42	12
T0477F06-RAD-RSS	A	1	1.17	38.47	255	1.30	40.04	208
	B	1	1.02	36.38	255	1.02	36.91	255
	A	2	2.82	47.83	101	2.14	46.94	93
	B	2	2.03	45.77	139	2.00	45.84	113
	A	4	4.28	61.05	18	4.05	59.45	22
	B	4	4.02	59.34	23	4.00	59.31	21

TABLE IV
RESULTS FOR SIMPLIFIED VARIANCE ESTIMATION.
LEFT: $N_e = N_{\text{columns}}/2$; RIGHT: $N_e = N_{\text{columns}}$

R_T	Rate (bpppb)	SNR (dB)	Rate (bpppb)	SNR (dB)
AGRICULTURE				
1	1.00	28.51	1.00	28.67
2	2.00	36.60	2.00	36.65
4	3.82	65.51	3.82	65.51
AIRS-GRAN9				
1	1.01	54.12	1.01	54.09
2	2.00	62.86	2.00	62.91
4	4.00	76.61	4.24	90.03
CRISM-SC214-NUC				
1	1.00	47.03	1.00	47.07
2	2.00	55.52	2.00	55.52
4	3.99	74.38	3.99	74.34

A. Bit-Plane Coding

A bit-plane coder subdivides its set of input samples in a number of binary sources, the so-called *bit planes*. Over a set of M consecutive samples of the mapped prediction residuals, each described with precision of D bits, a bit plane is the ordered sequence of all the M bits with the same significance. In the case of a hyperspectral image, the entropy coder will thus consist of a number of binary range coders, each one dedicated to a specific bit plane of the mapped prediction residuals [40], [41].

The implemented coder is based on a single binary range coder that exploits a binary model for the most significant bit plane (ms_bp) and two binary models for each of the other bit planes, i.e., $bp_0[b]$ and $bp_1[b]$ for bit plane b. Each binary model counts the occurrences of bits with value 0 and 1 in the appropriate bit plane. In particular, for models $bp_0[b]$ and $bp_1[b]$ the counts are conditioned to the value of the bit in the same position in the previous bit plane $b-1$, i.e., $bp_0[b]$ counts the numbers of zeros and ones in bit plane b when the colocated bit in bit plane $b-1$ is equal to zero, and analogously for $bp_1[b]$. This allows exploiting the correlation between adjacent bit planes, which would otherwise be lost. The statistical models are reset at the beginning of each new line of the image. The operations performed by the encoder are summarized in Algorithm 4. The bit-plane coding stage is fed with the value of the mapped quantized prediction residual of the current sample $m_{i,j,k}$, from which the current bit to be encoded ($current\_bit$) is extracted for the bit plane b currently being processed.

Algorithm 4 Bit-Plane Coding

1: **for** $k = 1 : N_{\text{bands}}$ **do**
2: Initialize binary models
3: **for** $i = 1 : N_{\text{columns}}$ **do**
Compute $m_{i,j,k}$
4: **for** $b = 0 : D-1$ **do**
Set $current\_bit = m_{i,j,k} \gg ((D-b-1))\ \&\ 0 \times 01)$
5: **if** $b == 0$ **then**
Encode $current\_bit$ with ms_bp
6: **else if** $m_{i,j,k} \gg ((D-b))\ \&\ 0 \times 01$ **then**
Encode $current\_bit$ with $bp_1[b]$
7: **else**

TABLE V
COMPARISON OF ENTROPY CODERS IN MODE B

Image	Target	Original rate	Original SNR	Bit-Plane rate	Bit-Plane SNR	Golomb + Range rate	Golomb + Range SNR
AGRICULTURE	0.25	0.26	12.31	0.27	18.40	0.29	10.95
	0.50	0.51	22.46	0.51	23.47	0.51	22.97
	1	1.01	28.54	1.01	29.59	1.01	29.21
	2	2.01	36.61	2.01	37.17	2.01	37.12
	4	3.82	66.38	3.67	inf	3.68	inf
AIRS-GRAN9	0.25	0.28	36.22	0.28	36.38	0.28	35.07
	0.50	0.50	46.59	0.50	46.82	0.49	46.16
	1	1.00	53.76	1.00	53.91	1.01	53.48
	2	2.04	63.02	2.02	63.21	2.05	62.93
	4	4.05	76.93	4.04	77.37	4.06	76.83
AVIRIS-SC0	0.25	0.36	32.33	0.35	32.34	0.38	32.33
	0.50	0.54	35.65	0.53	36.20	0.52	34.87
	1	1.02	45.71	1.01	46.42	1.02	45.83
	2	2.01	55.26	2.01	55.96	2.01	55.59
	4	4.00	69.02	4.01	69.57	4.01	69.30
CASI-T0477F06-NUC	0.25	0.38	19.75	0.37	19.90	0.61	19.82
	0.50	0.53	23.01	0.52	23.79	0.66	23.26
	1	1.01	38.12	1.01	40.53	1.01	39.03
	2	2.01	49.45	2.01	50.61	2.01	50.00
	4	4.01	62.89	4.01	63.21	4.01	62.92
CRISM-SC214-NUC	0.25	0.27	36.46	0.27	37.45	0.27	36.83
	0.50	0.52	41.27	0.52	42.24	0.52	41.56
	1	1.01	46.84	1.01	47.83	1.01	47.41
	2	2.01	55.50	2.01	56.38	2.01	56.21
	4	4.00	74.52	3.90	97.49	3.92	97.49
EO-1-GEO_SAMPLE-NUC	0.25	0.26	36.76	0.26	36.98	0.26	36.47
	0.50	0.51	40.91	0.51	41.17	0.51	40.72
	1	1.01	45.18	1.01	45.50	1.01	45.04
	2	2.01	52.67	2.01	53.04	2.01	52.54
	4	4.01	65.63	4.01	66.49	4.01	65.87
FRT00009326-07-VNIR	0.25	0.27	31.79	0.27	31.91	0.27	31.49
	0.50	0.52	34.66	0.52	34.75	0.52	34.36
	1	1.01	39.41	1.01	39.68	1.01	38.98
	2	2.01	48.14	2.01	48.91	2.01	48.34
	4	4.01	60.30	4.01	60.74	4.01	60.67
MODIS-MOD01-250m	0.25	0.26	18.40	0.26	19.63	0.26	18.88
	0.50	0.52	23.37	0.52	24.74	0.52	24.07
	1	1.02	30.01	1.02	31.60	1.02	31.12
	2	2.01	39.54	2.01	41.13	2.01	41.01
	4	4.01	53.69	4.01	55.58	4.01	55.31
MONTPELLIER	0.25	0.47	20.39	0.41	20.43	0.45	20.39
	0.50	0.51	20.64	0.52	21.92	0.51	20.98
	1	1.03	26.69	1.02	28.74	1.02	28.16
	2	2.09	35.45	2.03	37.27	2.03	37.08
	4	4.09	49.79	4.02	51.29	4.02	51.27
T0477F06-RAD-RSS	0.25	0.65	33.58	0.62	33.57	0.67	33.57
	0.50	0.68	33.71	0.65	33.77	0.68	33.62
	1	1.02	36.38	1.03	37.16	1.17	38.47
	2	2.03	45.77	2.02	46.63	2.22	47.83
	4	4.02	59.34	4.01	60.25	4.18	61.05

Encode $current\_bit$ with $bp_0[b]$
end if
end for
end for
end for

B. Cascaded Golomb + Range Encoder

In an alternative approach, a cascaded combination of a Golomb coder and a binary range coder is employed. The encoding is first done with the sample-adaptive Golomb coder used in the CCSDS-123 standard. The Golomb coder outputs its codewords into a temporary array every time an output byte is completed. The array is periodically checked for the presence of a new byte: If so, the eight bits are taken as a serial input for a binary adaptive range coder.

C. Results

Both the bit-plane range encoder and the cascaded Golomb/range encoder have been tested over the whole set of test images. The results provided by both algorithms in the rate control mode B are shown in Table V, along with a comparison with those obtained using the original encoder [1] in mode B.

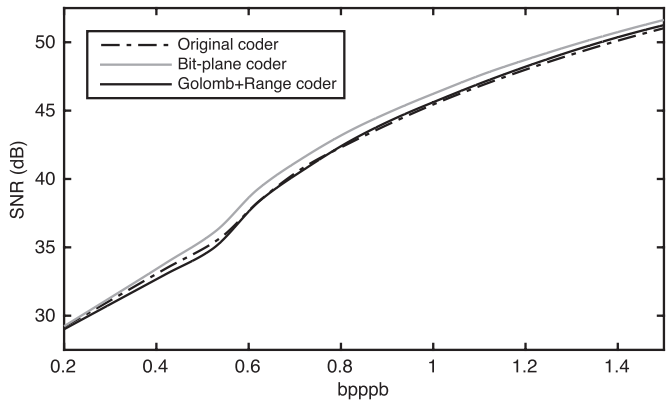

Fig. 4. Entropy Coders—Comparison for AVIRIS-SC0 Rate Control in Mode B.

For what concerns near-lossless coding, the bit-plane encoder has shown the best performance (i.e., a smaller bitrate for the same quality) for a vast majority of the images of the set. The use of binary sources makes the probability models easy to train. With respect to the original range encoder, the bit-plane-based one has a short adaptation time and almost always works with up-to-date probabilities. In our tests, this effect outweighed the loss of performance from neglecting bit-plane correlations, making the bit-plane encoder the most effective solution in terms of both performance and speed. The cascaded Golomb/range coder has shown competitive performance for some images of the set; it has however provided worse results in some cases (e.g., on the image AIRS-GRAN9) with respect to both the bit-plane coder and the original range coder, especially when very small bit rates are requested.

With very few exceptions, the bit-plane coder has shown to be preferable if rate control is desired: In Mode A, given a fixed SNR, it provides a lower bit rate; in Mode B, it achieves a smaller deviation from the desired target, still obtaining good results in terms of distortion. The Golomb + Range combined coder is not always preferrable to the other two ones if a rate control is needed: its performance is poorer at low bit rates, and it never outperforms the bit plane coder.

Fig. 4 shows a comparison between the rate–distortion curves obtained by the three algorithms, when applied to the image AVIRIS-SC0 in rate control Mode B.

VI. Conclusion

In this paper, we have presented several improvements over basic predictive lossy compression schemes. First, we have proposed an algorithm that allows performing lossy compression with bounded relative error between any pixel of the original and reconstructed images. The algorithm is very simple, and it allows approximately or exactly enforcing the user-provided bound on relative error. Second, we have proposed a simple and effective rate control algorithm, which improves significantly on the state of the art, by reducing the complexity and improving the performance with respect to [1]. Third, we have investigated several solutions for entropy coding, demonstrating the advantages of bit-plane-based entropy coding with respect to other solutions. These improvements make predictive lossy compression a very attractive solution for onboard compression in terms of both complexity and performance.

References

[1] D. Valsesia and E. Magli, "A novel rate control algorithm for onboard predictive coding of multispectral and hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6341–6355, Oct. 2014.

[2] A. B. Kiely and M. A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2672–2678, Aug. 2009.

[3] A. Abrardo, M. Barni, E. Magli, and F. Nencini, "Error-resilient and low-complexity onboard lossless compression of hyperspectral images by means of distributed source coding," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 4, pp. 1892–1904, Apr. 2010.

[4] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer, "Low-complexity lossless compression of hyperspectral imagery via linear prediction," *IEEE Signal Process. Lett.*, vol. 12, no. 2, pp. 138–141, Feb. 2005.

[5] B. Aiazzi, P. Alba, L. Alparone, and S. Baronti, "Lossless compression of multi/hyperspectral imagery based on a 3-D fuzzy prediction," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 5, pp. 2287–2294, Sep. 1999.

[6] B. Aiazzi, L. Alparone, and S. Baronti, "Near-lossless compression of 3-D optical data," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 11, pp. 2547–2557, Nov. 2001.

[7] B. Aiazzi, L. Alparone, S. Baronti, and C. Lastri, "Crisp and fuzzy adaptive spectral predictions for lossless and near-lossless compression of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 532–536, Oct. 2007.

[8] E. Magli, "Multiband lossless compression of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1168–1178, Apr. 2009.

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

[10] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.

[11] S. H. Cho and V. J. Mathews, "Tracking analysis of the sign algorithm in nonstationary environments," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 38, no. 12, pp. 2046–2057, Dec. 1990.

[12] M. Klimesh, "Low-complexity lossless compression of hyperspectral imagery via adaptive filtering," in *Proc. Interplanetary Netw. Progress Rep.*, 2005, pp. 42–163.

[13] Consultative Committee for Space Data Systems (CCSDS), "Lossless Multispectral and Hyperspectral Image Compression," *Blue Book*, no. 1, Recommended Std. CCSDS 123.0-B-1, May 2012. [Online]. Available: http://public.ccsds.org/publications/archive/123x0b1ec1.pdf

[14] S. Jain and D. Adjeroh, "Edge-based prediction for lossless compression of hyperspectral images," in *Proc. IEEE DCC*, 2007, pp. 153–162.

[15] J. Mielikainen and P. Toivanen, "Clustered DPCM for the lossless compression of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 12, pp. 2943–2946, Dec. 2003.

[16] J. Wu, W. Kong, J. Mielikainen, and B. Huang, "Lossless compression of hyperspectral imagery via clustered differential pulse code modulation with removal of local spectral outliers," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2194–2198, Dec. 2015.

[17] M. Ryan and J. Arnold, "The lossless compression of AVIRIS images by vector quantization," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 3, pp. 546–550, May 1997.

[18] J. Zhang, H. Li, and C. W. Chen, "Distributed lossless coding techniques for hyperspectral images," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 977–989, Sep. 2015.

[19] A. S. Mamatha, V. Kusuma, V. Singh, and M. P. R. Kumar, "Low complexity distributed approach to hyperspectral image compression," in *Proc. IEEE Region 10 Conf. TENCON*, Nov. 2015, pp. 1–6.

[20] D. S. Taubman, M. W. Marcellin, and M. Rabbani, "JPEG2000: Image compression fundamentals, standards and practice," *J. Electron. Imag.*, vol. 11, no. 2, pp. 286–287, 2002.

[21] *Document ISO/IEC 15444-2, JPEG 2000 Part 2—Extensions*. [Online]. Available: http://www.jpeg.org/metadata/15444-2.PDF

[22] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1408–1421, May 2007.

[23] I. Blanes and J. Serra-Sagristà, "Pairwise orthogonal transform for spectral image coding," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 961–972, Mar. 2011.

[24] I. Blanes, M. Hernández-Cabronero, F. Aulì-Llinàs, J. Serra-Sagristà, and M. W. Marcellin, "Isorange pairwise orthogonal transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3361–3372, Jun. 2015.

[25] K.-J. Cheng and J. Dill, "Lossless to lossy dual-tree BEZW compression for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5765–5770, Sep. 2014.

[26] Q. Du, N. Ly, and J. E. Fowler, "An operational approach to PCA+ JPEG2000 compression of hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2237–2245, Jun. 2014.
[27] L.-S. Lan and I. S. Reed, "Fast approximate Karhunen-Loève transform with applications to digital image coding," in *Proc. Vis. Commun.*, 1993, pp. 444–455.
[28] A. Pirooz and I. S. Reed, "A new approximate Karhunen–Loeve transform for data compression," in *Proc. Conf. Rec. 32nd Asilomar Conf. Signals, Syst. Comput.*, 1998, vol. 2, pp. 1471–1475.
[29] C. Lee, S. Youn, T. Jeong, E. Lee, and J. Serra-Sagristà, "Hybrid compression of hyperspectral images based on PCA with pre-encoding discriminant information," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 7, pp. 1491–1495, Jul. 2015.
[30] G. Carvajal, B. Penna, and E. Magli, "Unified lossy and near-lossless hyperspectral image compression based on JPEG 2000," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 593–597, Oct. 2008.
[31] J. Beerten, I. Blanes, and J. Serra-Sagristà, "A fully embedded two-stage coder for hyperspectral near-lossless compression," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 8, pp. 1775–1779, Aug. 2015.
[32] Z. He and S. Mitra, "Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 840–849, Oct. 2002.
[33] D. Valsesia and E. Magli, "A hardware-friendly architecture for on-board rate-controlled predictive coding of hyperspectral and multispectral images," in *Proc. IEEE ICIP*, 2014, pp. 5142–5146.
[34] R. Camarero, F. Dupuy, C. Thiebaut, C. Latry, and J.-M. Delvit, "Innovative techniques in predictive lossy compression for future CNES missions," in *Proc. OBPDC Workshop*, 2014, Art. no. 950102.
[35] G. N. N. Martin, "Range encoding: an algorithm for removing redundancy from a digitised message," in *Proc. Video Data Record. Conf.*, 1979, pp. 24–27.
[36] J. Mielikainen, "Lossless compression of hyperspectral images using look-up tables," *IEEE Signal Process. Lett.*, vol. 13, no. 3, pp. 157–160, Mar. 2006.
[37] M. De Nino, G. Capuano, M. Romano, and E. Magli, "Lossy multi/hyperspectral compression HW implementation at high data rate," in *Proc. Int. Astronaut. Congr.*, 2014, pp. 1–6.
[38] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
[39] I. Blanes, E. Magli, and J. Serra-Sagristà, "A tutorial on image compression for optical space imaging systems," *IEEE Geosci. Remote Sens. Mag.*, vol. 2, no. 3, pp. 8–26, Sep. 2014.
[40] J. Wu, M. Wang, Y. Fang, J. Jeong, and L. Jio, "Hyperspectral image compression using distributed arithmetic coding and bit plane coding," in *Proc. SPIE, Satellite Data Compress., Commun., Process. VI*, 2010, vol. 7810, Art. no. 781018.
[41] J. Schwartz and R. Barker, "Bit-plane encoding: A technique for source encoding," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-2, no. 4, pp. 385–392, Jul. 1966.

Marco Conoscenti received the Master's degree in computer engineering from Politecnico di Torino, Turin, Italy, in October 2015. His thesis concerned the analysis and development of rate control algorithms for hyperspectral image compression. He is currently working toward the Ph.D. degree in computer engineering with Nexa Center for Internet and Society, Department of Control and Computer Engineering, Politecnico di Torino.

His research interests include peer-to-peer networks and the Internet of Things.

Riccardo Coppola received the M.Sc. degree in computer engineering from Politecnico di Torino, Turin, Italy, in October 2015. He is currently working toward the Ph.D. degree in computer engineering with Politecnico di Torino.

His research interests include software engineering and testing.

Enrico Magli (S'97–M'01–SM'07) received the M.Sc. and Ph.D. degrees from Politecnico di Torino, Turin, Italy, in 1997 and 2001, respectively.

He is currently an Associate Professor with Politecnico di Torino. His research interests include compressive sensing, image and video coding, and vision.

Dr. Magli currently serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and the *EURASIP Journal on Image and Video Processing*. He served as an IEEE Distinguished Lecturer from 2015 to 2016. He received the 2010 and 2014 Best Associate Editor Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE Geoscience and Remote Sensing Society 2011 Transactions Prize Paper Award, and the IEEE ICIP 2015 Best Student Paper Award (as a Senior Author).