A Novel Rate-Controlled Predictive Coding Algorithm for Onboard Compression of Multispectral and Hyperspectral Images

A Novel Rate-Controlled Predictive Coding Algorithm for Onboard Compression of Multispectral and Hyperspectral Images

Diego Valsesia[(1)], Enrico Magli[(1)], and Maurizio De Nino[(2)]

[(1)] *Dipartimento di Elettronica e Telecomunicazioni - Politecnico di Torino*
Corso Duca degli Abruzzi 24 - 10129 Torino - Italy
diego.valsesia@polito.it, enrico.magli@polito.it
[(2)] *Techno System Developments*
Via Provinciale Pianura, 2 - 80078 Pozzuoli (NA) - Italy
mdenino@tsd-space.it

ABSTRACT

Predictive compression has always been considered an attractive solution for onboard compression thanks to its low computational demands and the ability to accurately control quality on a pixel-by-pixel basis. Traditionally, predictive compression focused on the lossless and near-lossless modes of operation where the maximum error can be bounded but the rate of the compressed image is variable. Fixed-rate is considered a challenging problem due to the dependencies between quantization and prediction in the feedback loop, and the lack of a signal representation that packs the signals energy into few coefficients as in the case of transform coding. In this paper, we show how it is possible to design a rate control algorithm suitable for onboard implementation by providing a general framework to select quantizers in each spatial and spectral region of the image and optimize the choice so that the desired rate is achieved with the best quality. In order to make the computational complexity suitable for onboard implementation, models are used to predict the rate-distortion characteristics of the prediction residuals in each image block. Such models are trained on-the-fly during the execution and small deviations in the output rate due to unmodeled behavior are automatically corrected as new data are acquired. The coupling of predictive coding and rate control allows the design of a single compression algorithm able to manage multiple encoding objectives. We tailor the proposed rate controller to the predictor defined by the CCSDS-123 lossless compression recommendation and study a new entropy coding stage based on the range coder in order to achieve an extension of the standard capable of managing all the following encoding objectives: lossless, variable-rate near-lossless (bounded maximum error), fixed-rate lossy (minimum average error), and any in-between case such as fixed-rate coding with a constraint on the maximum error. We show the performance of the proposed architecture on the CCSDS reference dataset for multispectral and hyperspectral image compression and compare it with state-of-the-art techniques based on transform coding such as the use of the CCSDS-122 Discrete Wavelet Transform encoder paired with the Pairwise Orthogonal Transform working in the spectral dimension. Remarkable results are observed by providing superior image quality both in terms of higher SNR and lower maximum error with respect to state-of-the-art transform coding.

INTRODUCTION

Image spectrometers collect vast amounts of data which can be used for a variety of tasks. Fine spectral resolution can be a desired featured when it comes to detecting fingerprints in the spectral response of a scene, but a problem of handling such wealth of information naturally arises and calls for the use of compression methods. Onboard compression enables spacecrafts to save transmission time, allowing more images to be sent to the ground stations. The design of compression algorithms for onboard applications must carefully meet the limited resources in terms of computational power and memory available on the spacecrafts. Two main compression techniques are available in this scenario: transform coding and predictive coding.

Transform coding relies on computing a linear transform of the data to achieve energy compaction and hence transmit few carefully chosen transform coefficients. One of the most popular approaches is JPEG2000 [1] and its multidimensional extension. A wavelet-based 2D lossless and lossy compression algorithm has also been standardized for space applications [2]. Spectral transforms to eliminate the inter-band redundancy have been subject of intense research. There exists an optimal transform for Gaussian sources, *i.e.*, the Karhunen-Loève transform (KLT) but its complexity does not match the computational resources typically available

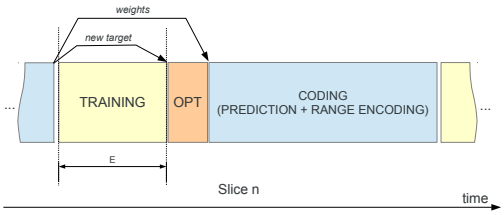

Figure 1. Serial rate control implementation.

for onboard compression. Hence, low-complexity approximations to the KLT have been derived, such as the Pairwise Orthogonal Transform (POT) [3]. Transform coding allows to perform lossless and lossy compression and to accurately control the rate in a simple manner thanks to the simple relation between rate and quantized transform coefficients [1]. On the other hand, per-pixel quality control as in near-lossless compression is hard to obtain. Transform coding also typically suffers from the problem of dynamic range expansion, which is a direct consequence of energy compaction. While it is difficult to generalize due to the availability of many different transforms and predictors, a transform generally uses many (past and future) pixels of the image to represent a given pixel, while a predictor generally employs few pixels in a causal neighborhood, thus making it less prone to performance loss when the prediction is reset over different image areas, *e.g.*, in order to achieve error resilience.

Predictive coding uses a mathematical model to predict pixel values and encode only the prediction error. Adaptive linear prediction is often used [4, 5, 6, 7] (*e.g.*, the predictor adopted by the CCSDS-123 reccomendation [8] relies on the LMS filter [9], with the sign algorithm [10] for weight update), but other methods have been devised as well, *e.g.*, based on edge detection [11] or vector quantization [12]. In lossless compression, the prediction residuals are written in the compressed file after entropy coding. Lossy compression instead quantizes them before entropy coding. The quantization step size determines the amount of compression and hence information losses with respect to the original image. Near-lossless compression is readily implemented by setting a maximum quantization step size, so that the quantization error never exceeds half of it. On the other hand, rate control in a predictive coder is challenging because: i) no simple mathematical relationship between the rate and the quantized prediction residual exists, ii) the quality of the prediction, hence the magnitude of the residuals, and ultimately the rate depend on how coarse the quantization is.

In this paper we review an innovative design of a rate controller for a predictive encoder, proposed in [13]. We show that the proposed method can achieve accurate control, while having complexity suitable for onboard implementation. In particular, the algorithm is designed to work in line-based acquisition mode, as this is the most typical setup of spectral imaging systems. We first describe the proposed algorithm in general terms, as it can be applied to any predictive coder but then, we focus our attention on using it with the LMS predictor used in the CCSDS-123 standard for lossless compression [8], which is an improved version of the Fast Lossless algorithm [14]. The resulting system can be seen as an extension of the standard featuring lossless, near-lossless and rate-controlled lossy compression. The rate controller provides lossy reconstructions with increasingly better quality, up to lossless encoding, as the target rate approaches that of lossless compression. The controller can also work in a hybrid rate-controlled and near-lossless mode by specifying the maximum quantization step size that the controller is allowed to use. Finally, we discuss the architecture proposed in [15] which overcomes the serial nature of the algorithm, as well as the large memory requirements of the entropy coding stage, achieving a pipelined implementation suitable for high-throughput onboard implementation, at a negligible cost in terms of coding efficiency.

RATE CONTROL ALGORITHM

The purpose of the algorithm is to control the output rate of a predictive encoder of hyperspectral and multispectral images, under low complexity and memory constraints. This rate control algorithm can work with any predictor, as it selects the quantizers operating on the prediction errors. The rate control algorithm works on a slice-by-slice basis, where we call "slice" a predefined number of lines with all their spectral channels. Each slice is divided into nonoverlapping 16×16 blocks. An individual quantization step size is computed for each block in each spectral channel, so that lossy predictive coding employing the computed step sizes will achieve a rate as close as possible to the target. The rate control algorithm is a two stage process that computes such step sizes, as depicted in Fig. 1. In particular the following steps are performed:

$$R(\Lambda, Q) = -\left(1 - e^{-\Lambda\frac{Q}{2}}\right)\log_2\left(1 - e^{-\Lambda\frac{Q}{2}}\right) - \frac{e^{-\Lambda\frac{Q}{2}}}{\log(2)}\left[\log\left(\frac{1 - e^{-\Lambda Q}}{2}\right) + \frac{\Lambda Q}{2} - \frac{\Lambda Q}{(1 - e^{-\Lambda Q})}\right] \quad (2)$$

$$D(\Lambda, Q) = \frac{2 - \frac{1}{4}e^{-\Lambda\frac{Q}{2}}\left(\Lambda^2 Q^2 + 4\Lambda Q + 8\right)}{\Lambda^2} + \frac{-\Lambda Q\left(\Lambda Q + 4\right) + e^{\Lambda Q}\left[\Lambda Q\left(\Lambda Q - 4\right) + 8\right] - 8}{4\Lambda^2}\frac{e^{-\frac{3}{2}\Lambda Q}}{1 - e^{-\Lambda Q}} \quad (3)$$

- *Training* stage: a model predicting the rate-distortion curve of each block in the slice is built as function of the variance of the unquantized prediction residuals and of the quantization step size. The former is estimated by running the lossless predictor on a small number of lines in the slice. The number of lines employed in the training is denoted as E. While the best choice in terms of estimation error is using all the lines in the slice, this operation is costly, so E is typically kept to a small value such as 2 lines only.
- *Optimization* stage: the final quantization step sizes are obtained as follows. First, an initial set of quantization step sizes is calculated, which approximately achieves the target rate but is suboptimal in terms of distortion. Then, a greedy algorithm makes local adjustments aimed at promoting low-distortion allocations of the quantization step sizes, employing the rate-distortion models of all blocks in the slice.

Furthermore, the algorithm measures the actual rate produced by encoding the slice with the computed quantization step sizes and uses this information to update the target rate for the next slices. This mode of operation has been shown to effectively correct inaccuracies in the model without reducing the rate-distortion performance.

Training

We now introduce the model used to describe the prediction residuals in each block. This model allows to obtain closed-form expressions for the rate and the distortion of the quantized residuals in the block. It is commonly observed that accurate predictors tend to yield residuals with leptokurtic (high kurtosis) distribution, hence similar to the Laplace probability density function, which we use to model the distribution of prediction residuals:

$$f_r(x) = \frac{\Lambda}{2}e^{-\Lambda|x|}, \quad (1)$$

where Λ is related to the variance σ^2 of the distribution by $\Lambda = \sqrt{\frac{2}{\sigma^2}}$. We assume that the residuals in each block and the blocks themselves are independent of each other. The residuals are quantized using a quantization step size Q. We can derive analytic expressions for the rate and the distortion (MSE) of the quantized residuals, which are (2) and (3). We can notice that both the rate and the distortion are functions of the variance σ^2 of the unquantized residuals in the block and of the quantization step size Q, whose value is yet unknown. Each block in the slice has its own variance parameter and quantizations step size. The variance must be estimated, while obtaining the quantization step size is really the ultimate goal of the rate control algorithm. Estimation of the variance is done by running the predictor without quantization of the residuals for E lines.

Optimization

Algorithm 1 Projection algorithm to solve (5)

Sort $R(\mathbf{\Lambda}, \mathbf{Q})$ into μ in descending order
Find $\rho = \max\left\{j : \mu_j - \frac{1}{j}\left(\sum_{i=1}^{N_B}\mu_i - R_{target}\right) > 0\right\}$
Define $\theta = \frac{1}{\rho}\left(\sum_{i=1}^{N_B}\mu_i - R_{target}\right)$
Find $\mathbf{w}$ such that $w_i = \max\left\{R(\Lambda_i, Q_i) - \theta, 0\right\}$
Find $\hat{\mathbf{Q}} = R^{-1}(\mathbf{\Lambda}, \mathbf{w})$

The final quantization step sizes are obtained after a sequence of two procedures called l_1 *projector* and *Selective Diet*. First, the integer problem of choosing the quantization step sizes is approximated by a continuous one and a projection step onto the simplex defined by the rate constraint is taken. Then, Selective Diet

Algorithm 2 Selective Diet

Require: $\mathbf{Q_g}$, $\lambda = 50$, N_{iter}
  for $iter = 1 \rightarrow N_{iter}$ **do**
    Set **default** = $\mathbf{Q_g}$, $\mathbf{Q}^{(+2)}$ = **Q+2**, $\mathbf{Q}^{(-2)}$ = **Q-2**
    Set output chain $\mathbf{Q_g} = \mathbf{Q}^{(-2)}$
    Compute $R_{diff} = \sum R(\mathbf{Q_g}) - R_{target}$, *i.e.*, the rate you need to lose to reach the target
    Sort the nodes in $\mathbf{Q}^{(+2)}$ by decreasing value of $J_i = \left(D_i^{(-2)} - D_i^{(+2)}\right) + \lambda \left(R_i^{(-2)} - R_i^{(+2)}\right)$
    $i = 1$
    while $\sum R(\mathbf{Q_g}) - R_{target} < R_{diff}$ **do**
      Replace the corresponding node in $\mathbf{Q_g}$ with the i-th node in the sorted $\mathbf{Q}^{(+2)}$
      $i = i + 1$
    end while
    if $iter \neq 1$ **then**
      if Distortion did not lower AND inner iterations not exceeded **then**
        Set $\lambda \leftarrow \lambda/2$ and repeat current iteration
      else
        Proceed to next iteration
      end if
    end if
  end for

performs an optimization of the solution returned by the projector by making local integer adjustments aimed at reaching the target rate and promoting low-distortion allocations of the quantization step sizes, according to the theoretical model. Some adjustments, described in [13], are made to this basic scheme in order to deal with dependencies among prediction residuals, originating in the propagation of quantization noise in the prediction loop and to deal with blocks where the estimated variance is deemed too low and unrealiable.

Concerning the l_1 projector, suppose that the encoder is given a target rate for the encoded image equal to T bits-per-pixel (bpp), and suppose that there are N_B blocks in the current slice (N_B is the product of the number of blocks in one band times the number of bands). We define the quantity $R_{\mathsf{target}} = T \cdot N_B$ as the product of the target rate in bpp and the number of blocks in the slice (note that this quantity does not represent the actual number of bits at our disposal since we are multiplying times the number of blocks and not the number of pixels). Ideally we would like to satisfy the rate constraint exactly, hence have

$$\sum_{i=1}^{N_B} R(\Lambda_i, Q_i) = R_{\mathsf{target}} \tag{4}$$

where Q_i is the quantization step size selected for the i-th block. Notice that since the rate of each block is a positive quantity, (4) defines a simplex in N_B dimensions. We can consider an initial solution having $Q_i = 1 \ \ \forall i$ (lossless encoding), with corresponding rates $R(\Lambda_i, 1)$. Geometrically, we have a vector in an N_B-dimensional space whose entries are the rates $R(\Lambda_i, 1)$ and we can project it onto the simplex defined by (4). In other words, we seek to solve the following optimization problem, where we slightly abuse notation using boldface to indicate N_B-dimensional vectors and making the R function operate component-wise:

$$\hat{\mathbf{R}} = \arg\min_{\mathbf{R}} \|\mathbf{R} - R(\mathbf{\Lambda}, \mathbf{1})\|_2 \quad \mathsf{subject\ to} \quad \|\mathbf{R}\|_1 = R_{\mathsf{target}} \tag{5}$$

Problem (5) is a continuous problem, whereas quantization step sizes are odd-integer-valued. After solving (5) we need to search the value of $\hat{Q}_i$ such that $R(\Lambda_i, \hat{Q}_i)$ is closest to $\hat{R}_i$. Any search method such as linear search or binary search can be used for this purpose. The algorithm is summarized in Algorithm 1. This solution, albeit inaccurate, is a good starting point to initialize the Selective Diet algorithm, summarized in Algorithm 2. Selective Diet is a local search method trying to solve an integer optimization problem consisting in lowering the distortion of the encoded slice while satisfying the constraint on its final rate. It does so by making local adjustments to the solution provided by the l_1 projector, hence the need for a good initialization point. For convenience of explanation, we shall represent the blocks in the current slice as nodes in a chain. It is possible to modify the chain by making adjustments to the nodes, namely changing the quantization step size assigned to that node. Only local adjustments are allowed: the quantization step of each node can only be increased by 2 or

decreased by 2. We shall call *+2 level* an assignment of $Q_i + 2$ where Q_i is the current value of the quantization step, called *default level*, and *-2 level* an assignment equal to $Q_i - 2$. A chain can be formed by choosing one of those three levels for each and every node. Consistently with the notation, we will call *+2/default/-2 chain* a chain made only of nodes in the +2/default/-2 level. The ultimate goal of Selective Diet is creating a chain that meets the rate constraint and has low distortion and to do so it moves some nodes to the -2 and +2 levels. The starting point is to consider the -2 chain as the new candidate output chain, since it has the lowest distortion. Obviously, selecting the -2 chain causes an increase in the rate, which must be compensated to meet the target. In order to reduce the rate moving back towards the target, some nodes are assigned to the +2 level. Each node is associated a cost function that considers the trade-off between the gain in rate reduction and the loss in quality due to switching from the -2 to the +2 level. The following cost function modelling the trade-off with a Lagrange multiplier is used:

$$J_i = \left[D(\Lambda_i, Q_i^{(-2)}) - D(\Lambda_i, Q_i^{(+2)})\right] + \lambda \left[R(\Lambda_i, Q_i^{(-2)}) - R(\Lambda_i, Q_i^{(+2)})\right] \qquad i \in [1, N_B] \tag{6}$$

The nodes are sorted by decreasing value of this cost function and this is the order in which the nodes are selected to be assigned to the +2 level. Specifically, one node at a time is added to the +2 level until the rate reaches R_{target}. The new chain is then formed by the nodes that remained at the -2 level and the nodes that were demoted to the +2 level. This chain is taken as the new default chain for a new iteration of the algorithm in order to try to further improve distortion. Notice that even if in a single iteration the algorithm selects nodes from the +2 and -2 levels only, it is possible to reach any value of Q using successive iterations, thus considering all possible odd values of the quantization step as possible choices for any block. The algorithm is run in a greedy manner, stopping when the distortion is not improving further. We have experimentally observed that the algorithm requires very few iterations (typically less than 10). Finally, the value of λ controls the tradeoff between the reduction in rate and increase in distortion when adding a node to the +2 level. The optimal value of λ would let us choose those nodes that allow a maximization of the gain in rate and a minimization of the increase in distortion. However, finding the optimal value would be computationally very demanding, so we resort to initializing λ to an empirically determined value ($\lambda = 50$) that we observed to be performing nicely over the whole test image set.

Rate feedback

The rate control algorithm outlined so far is completely model-based, meaning that no information about the real rate of the encoded slices is available. A more accurate control can be achieved by adding a feedback mechanism that modifies the target rate for future slices based on the actual rate used to encode the previous slices. Note that we do not want to increase the complexity of the system, hence we are not performing a multi-pass encoding of the same slice but rather correcting the target for future slices. The system adopts a Least-Mean-Square tracking approach to determine the target rate for the next slice, after measuring the rate produced by the encoding of the current slice. The target update formula is derived to take into account two issues. First, the inaccuracies in the rate controller make the actual output rate different from the target, thus we want to estimate the input-output relationship of the controller and track it in case of nonstationary behaviour. Second, we would like to count how many bits were used up to the current slice, and modify the target rate depending on the amount of bits that we saved, and we would like to spend on the next slices or, viceversa, the number of bits that we spent but we should have not. The goal is to try to assign all, but not more than the budget bits at our disposal, by spending them on or saving them from the remaining slices. The final rate update formula, to be motivated hereafter, is:

$$T_{new}[n+1] = \eta[n+1] + \frac{c[n+1]}{\tau} \cdot \frac{1}{\bar{w}[n]} \tag{7}$$

with

$$c[n+1] = \sum_{k=0}^{n} (T - y[k]) = c[n] + T - y[n] \tag{8}$$

$$\eta[n+1] = \eta[n] + \bar{w}[n] \left[T - y[n] + \frac{c[n]}{\tau}\right] \tag{9}$$

$$\bar{w}[n] = \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} w[k] \tag{10}$$

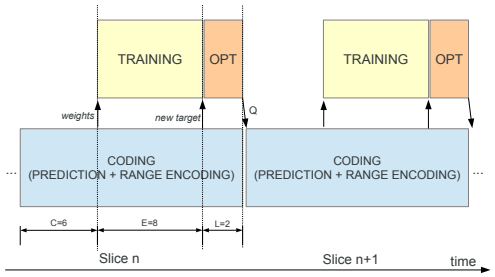

Figure 2. Parallel implementation.

where $y[n]$ is the actual rate produced encoding slice n, $T_{new}[n+1]$ is the target rate specified to the $(n+1)$-th slice, which is the next slice to be coded, and T is the original target rate for the whole image (and the initial condition for T_{new}). $c[n]$, which we call "residual budget", stores how much deviation in rate from T has been accumulated up to slice n. The τ factor used in the formulas plays the role of a time constant, ideally distributing the residual budget over τ future slices. $\mathcal{I}$ is the set of previous slices to be used.

PARALLEL ARCHITECTURE

The first issue concerning an efficient implementation of the rate control algorithm is its serial structure, as shown in Fig. 1, which requires to first perform the training stage. Only when this step is completed, the algorithm can move to the optimization stage. When this is done, the task of the rate controller is completed and the actual coding pass can begin, in which the residuals are quantized using the assigned step sizes. Conversely, designing a pipeline where the rate controller computes the quantization steps for a future $(n+1)$-th slice in parallel to the encoding of the current n-th slice, is highly desirable. Indeed, such a pipelined implementation would eliminate any throughput decrease due to the rate control operation, which could be executed by a software thread or hardware module of its own. Moreover, additional benefits would include the ability of using a larger value of E, thus providing a more accurate estimate of the variance of unquantized prediction residuals due to the availability of more lines for this task.

In order to accomplish this objective, two main obstacles have to be faced: i) the need to provide an updated predictor for the training phase (*e.g.*, updated weights of an LMS filter), and ii) the need to provide an updated target rate for slice $n+1$ to the rate controller based on the actual rate produced for slice n. The former problem can be solved by allowing the thread coding slice n to complete a certain number of lines (C) before passing the predictor parameters to the rate control thread, which is ready to start variance estimation for slice $n+1$. Avoiding to perform a full weight adaptation over $C+E$ lines allows to anticipate the beginning of the rate control task for slice $n+1$, which can finish before the beginning of the coding task for slice $n+1$, allowing to pipeline the rate control and coding stages. The latter issue is solved by noticing that the target rate is needed only by the optimization stage. Thus, in the proposed architecture the coding thread passes the information about the actual rate only after the rate controller has completed the training phase. This allows the coding thread to complete a significant number of lines (about $C+E$ lines), so that the actual output rate reliably represents the rate of the full slice, and can thus be used to update the target rate for future slices effectively. Synchronization can be easily managed by forcing the coding thread to pass the rate information when L lines are missing before completing the coding of slice n. We can notice that most of the complexity of the rate controller lies in the training phase and not in the optimization phase, so that L can have a small value. The rate control thread returns the assigned quantization step sizes, just in time for the coding thread to start processing slice $n+1$. Fig. 2 shows the interactions between the coding and rate control threads. It is worth noticing that several tradeoffs are available for different choices of the parameters. A typical setup uses slices of 16 lines, with $C=6$, $E=8$, and $L=2$.

RANGE CODER

The CCSDS-123 recommendation defines an adaptive coding approach using Golomb-Power-of-2 codes, mainly due to its low complexity and good performance, as well as the existence of an earlier standard (CCSDS 121.0-B [16]) using the Rice coding algorithm, embedded in the block-adaptive mode. We propose a different entropy coding stage based on the range coder [17]. The range coder is essentially a simplified arithmetic encoder.

Such a block coder is needed in order to achieve rates lower than 1 bpp, as the minimum codeword length for the Golomb code is 1 bit. Moreover, a higher performance entropy coder improves the effectiveness of the rate controller, by limiting the suboptimality introduced at this stage. An earlier version of the proposed range coder kept four separate models for each band for the prediction residuals, as described in [18]. Some modifications to this structure have been made to address the excessive memory consumption. First, prediction residuals are mapped onto non-negative integers following the same scheme adopted in CCSDS-123, thus the sign model only has to distinguish a zero residual from a positive one. Then, a total of four models are kept for all the spectral channels, i.e., the same structure storing the model is updated during the coding process, which follows a Band-Interleaved-by-Line (BIL) order. The proposed approach is suboptimal with respect to the original solution because a single model cannot discriminate the different statistics of the prediction residuals in the various bands. However, this scheme allows to reduce the memory requirement by factor equal to the number of bands, which can be very large. In [15] we showed that this structure incurs in a very small performance loss in terms of output rate.

NUMERICAL RESULTS

We have performed extensive tests on images extracted from the corpus defined by the MHDC working group of the CCSDS for performance evaluation and testing of compression algorithms. For brevity, we report the results of a comparison between the performance of the proposed extension of CCSDS-123 to lossy compression with rate control against a state-of-the-art transform coder intended for onboard compression. The CCSDS-122 standard [2] defines a transform coder employing the Discrete Wavelet Transform and a low-complexity Bit Plane Encoder, for the compression of 2D imagery. An extension of such standard to multiband imagery by including a spectral transform has been implemented and is publicly available online [19]. The implementation combines the CCSDS-122 encoder with the POT spectral transform [3]. The proposed system is instead run using the rate control algorithm with slice-by-slice rate feedback with $\tau = 5$, and memory-1. Full prediction mode and neighbor-oriented local sums are the parameters of CCSDS-123 predictor. We remark that the availability of the rate controller for the predictive system allows to perform a direct comparison, in which both systems work in a pure rate-controlled fashion by specifying a target rate and letting the encoder perform all the coding decisions automatically. The proposed rate controller is operated using $\mathrm{E} = 8$ for both the serial and parallel version. Table 1 reports a comparison between the proposed system with serial an parallel architectures and the transform coding method. The proposed predictive system is competitive against transform coding by typically providing superior quality, both in terms of SNR and in terms of maximum absolute distortion (MAD), for the same rate. Other quality metrics such as the maximum spectral angle (MSA) and average spectral angle (ASA) have been studied in the literature, but we omit them for reasons of brevity. However, such metrics follow the same trends observed for SNR and MAD, respectively. We observe that, at lower rates, the proposed algorithm achieves significant gains in terms of MAD even when the SNR gain is small or for the few cases when the transform coder is more effective.

Table 1. Performance comparison.

	PREDICTIVE (parallel)			PREDICTIVE (serial)			TRANSFORM	
IMAGE	RATE (bpp)	SNR (dB)	MAD	RATE (bpp)	SNR (dB)	MAD	SNR (dB)	MAD
AIRS_GRAN9	2.015	63.19	4	1.988	63.13	4	60.76	19
$135 \times 90 \times 1501$	4.034	77.16	1	3.986	76.70	1	70.42	4
AVIRIS_SC0	1.998	55.84	24	2.001	56.07	24	55.02	107
$512 \times 680 \times 224$	4.001	69.39	3	3.993	69.52	3	65.03	21
CRISM-SC214-NUC	2.000	56.35	9	1.935	56.32	4	52.72	45
$510 \times 640 \times 545$	3.955	94.07	1	3.819	97.03	1	65.32	3
MODIS-MOD01_500M	2.001	39.10	87	2.009	39.41	90	36.54	244
$4060 \times 2708 \times 5$	4.001	53.70	12	4.005	54.18	12	49.77	53
MONTPELLIER	2.030	36.78	47	2.025	37.14	43	33.46	635
$224 \times 2456 \times 4$	4.035	50.78	8	4.020	51.15	7	45.44	47
VGT1_1B	2.000	39.84	31	2.004	40.22	28	37.05	231
$10080 \times 1728 \times 4$	4.003	53.25	5	4.000	53.60	5	49.76	15

References

[1] David S Taubman, Michael W Marcellin, and Majid Rabbani, "JPEG2000: Image compression fundamentals, standards and practice," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 286–287, 2002.

[2] Consulative Committee for Space Data Systems (CCSDS), "Image Data Compression," *Blue Book*, November 2005.

[3] Ian Blanes and Joan Serra-Sagristà, "Pairwise orthogonal transform for spectral image coding," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 3, pp. 961–972, 2011.

[4] Aaron B Kiely and Matthew A Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 8, pp. 2672–2678, 2009.

[5] Andrea Abrardo, Mauro Barni, Enrico Magli, and Filippo Nencini, "Error-resilient and low-complexity onboard lossless compression of hyperspectral images by means of distributed source coding," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 4, pp. 1892–1904, 2010.

[6] Francesco Rizzo, Bruno Carpentieri, Giovanni Motta, and Jame A Storer, "Low-complexity lossless compression of hyperspectral imagery via linear prediction," *Signal Processing Letters, IEEE*, vol. 12, no. 2, pp. 138–141, 2005.

[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 Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 532–536, Oct. 2007.

[8] Consulative Committee for Space Data Systems (CCSDS), "Lossless Multispectral and Hyperspectral Image Compression," *Blue Book*, , no. 1, May 2012.

[9] Bernard Widrow, John M McCool, Michael G Larimore, and C Richard Johnson Jr, "Stationary and nonstationary learning characteristics of the lms adaptive filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.

[10] Snng Ho Cho and V John Mathews, "Tracking analysis of the sign algorithm in nonstationary environments," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 12, pp. 2046–2057, 1990.

[11] S.K. Jain and D.A. Adjeroh, "Edge-based prediction for lossless compression of hyperspectral images," in *Data Compression Conference, 2007. DCC '07*, 2007, pp. 153–162.

[12] M.J. Ryan and J.F. Arnold, "The lossless compression of aviris images by vector quantization," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 35, no. 3, pp. 546–550, 1997.

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

[14] Matthew A Klimesh, "Low-complexity lossless compression of hyperspectral imagery via adaptive filtering," 2005.

[15] D. Valsesia and E. Magli, "A hardware-friendly architecture for onboard rate-controlled predictive coding of hyperspectral and multispectral images," in *Image Processing (ICIP), 2014 IEEE International Conference on*, October 2014.

[16] Consulative Committee for Space Data Systems (CCSDS), "Lossless Data Compression," *Blue Book*, May 2012.

[17] G Nigel N Martin, "Range encoding: an algorithm for removing redundancy from a digitised message," in *Video and Data Recording Conference*, 1979, pp. 24–27.

[18] J. Mielikainen, "Lossless compression of hyperspectral images using lookup tables," *Signal Processing Letters, IEEE*, vol. 13, no. 3, pp. 157–160, 2006.

[19] Group on Interactive Coding of Images, "Delta software (a futurible CCSDS implementation)," 2013.