POLITECNICO DI TORINO
Repository ISTITUZIONALE

A hardware-friendly architecture for onboard rate-controlled predictive coding of hyperspectral and multispectral images

Original
A hardware-friendly architecture for onboard rate-controlled predictive coding of hyperspectral and multispectral images / Valsesia, Diego; Magli, Enrico. - (2014), pp. 5142-5146. (Intervento presentato al convegno 2014 IEEE International Conference on Image Processing (ICIP)) [10.1109/ICIP.2014.7026041].

Availability:
This version is available at: 11583/2592612 since:

Publisher:
IEEE - INST ELECTRICAL ELECTRONICS ENGINEERS INC

Published
DOI:10.1109/ICIP.2014.7026041

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

A HARDWARE-FRIENDLY ARCHITECTURE FOR ONBOARD RATE-CONTROLLED PREDICTIVE CODING OF HYPERSPECTRAL AND MULTISPECTRAL IMAGES

Diego Valsesia *Enrico Magli*

Politecnico di Torino (Italy)
Department of Electronics and Telecommunications
{diego.valsesia,enrico.magli}@polito.it

ABSTRACT

In this paper we propose an efficient architecture for onboard implementation of rate-controlled predictive lossy compression of hyperspectral and multispectral images. In particular, we consider the recent state-of-the-art rate control algorithm for onboard predictive compression [1], and propose an architecture addressing two fundamental aspects of its hardware implementation. Specifically, this architecture 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.

***Index Terms*—** Hyperspectral image coding, embedded systems, predictive coding, rate control

1. INTRODUCTION

Imaging spectroscopy allows to capture information about a scene at several different wavelengths and encode it in a multispectral or hyperspectral image. Such images provide a great wealth of information and are useful for a variety of tasks ranging from terrain analysis to military surveillance. It often happens that this kind of imaging is performed onboard of spacecrafts, thus handling so much data can be a challenging task. In particular, we address the compression problem, where the amount of data to transmit to ground stations must be reduced, whereas the scarce computational resources available onboard call for low-complexity techniques.

Extensive literature is available on the subject of lossless and lossy compression but two main approaches can be identified: transform coding and predictive coding. The former is based on the use of a linear transform to change the signal representation into a domain where it has a very compact representation. It has been very successful for coding of 2D images as proved by the JPEG and JPEG 2000 standards, as well as the CCSDS-122 recommendation for space systems [2], and for coding of multidimensional images where a spectral transform [3, 4] can be used to eliminate inter-band redundancy. Predictive coding [5, 6, 7, 8] relies on the use of a mathematical model to predict pixel values and encode the prediction error only. It typically presents low memory requirements and few operations needed to perform prediction, which are highly desirable features for onboard compression algorithms. In a scenario of fixed rate, space applications typically operate transform encoders in a rate-controlled fashion, *i.e.*, by specifying the desired target rate, while predictive coders focus on lossless and near-lossless (*i.e.*, bounded l_∞ error) compression. The reason is that while rate control is naturally obtained for transform coding [9], it is challenging for predictive coding, and vice versa concerning bounding the maximum error.

In a recent work, Valsesia *et al.* [1] proposed a rate control algorithm for onboard predictive coding. The proposed algorithm has been used together with the predictor adopted in the CCSDS-123 lossless compression recommendation [10], to develop a unified tool performing lossless, near-lossless and rate-controlled lossy compression, as well as hybrid rate-controlled compression with a near-lossless constraint, providing superior performance with respect to state-of-the-art transform coding approaches to onboard compression. Despite the rate control algorithm has indeed low computational complexity, it requires a serial hardware implementation in which first a few spectral lines of a hyperspectral image are encoded, and only when this process is finished it is possible to start the rate-distortion optimization process for the next spectral lines. This is due to the fact that the previously encoded lines are employed to estimate the operational rate-distortion curve of the next lines, which is in turn used to select appropriate quantization step sizes that will satisfy the rate constraint. However, the serial nature of the algorithm unnecessarily decreases the achievable hardware throughput. Moreover, to achieve improved coding efficiency, as well as to enable rates below 1 bpp, a simplified arithmetic coder is used, *i.e.*, a range coder. This raises the problem of the statistical model, which usually occupies a very large amount of memory that is not available on-chip on typical FPGAs for space applications.

In this paper, we present a new architecture that employs the rate control algorithm of [1] to achieve an efficient pipelined implementation without compromising the cod-

This work was supported by the European Space Agency (ESA-ESTEC) under grant 107104.

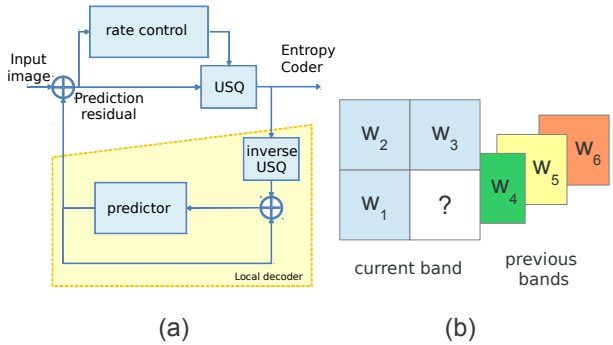

Fig. 1. (a) Predictive coder, (b) Example of prediction context.

ing efficiency. Moreover, we show that the statistical model employed by the entropy coding stage can be simplified to significantly reduce the amount of required memory, thus solving one of the main issues regarding its implementation.

2. BACKGROUND

This section reviews the rate control algorithm originally proposed in [1]. In particular, we focus on the intuition behind the algorithm, while all details can be found in [1]. 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. Fig. 1a shows a high-level description of a prediction-based encoder with a rate controller block adjusting the uniform scalar quantizer (USQ). An example of predictor is a Least Mean Square (LMS) filter predicting a pixel value as a weighted sum of pixels in a causal spatial and spectral neighbourhood of the current pixel, as shown in Fig. 1b. The weights w_1, w_2, ..., w_6 are updated at each pixel, applying gradient techniques to minimize the prediction error [11]. In [1] the rate controller has been tested with the predictor defined by the CCSDS-123 recommendation, which is basically an LMS filter using the low-complexity sign algorithm [12] for weight update.

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. 2. In particular the following steps are performed:

- *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.

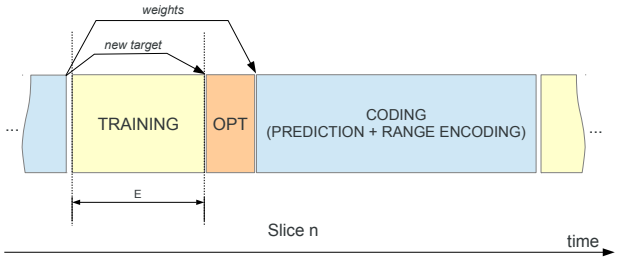

Fig. 2. Serial implementation.

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.

3. PROPOSED ARCHITECTURE

3.1. Parallel rate control

The first issue concerning an efficient implementation of the rate control algorithm is its serial structure, as shown in Fig. 2, which requires to first perform the training stage. In this stage, the variance of the unquantized prediction residuals is estimated by running the predictor without quantization of the residuals on the first E lines of the slice, for all the spectral channels. Only when this step is completed and each block has been associated with the corresponding value of variance of its prediction residuals, the algorithm can move to the optimization stage where the quantization step sizes are allocated by invoking the rate-distortion optimization procedures. 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. 3 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 $L=2$. The following behavior is associated to each of the parameters:

- C (*Coded* lines): number of lines before the rate control is started. The higher it is, the more the predictor in the rate control thread will be up-to-date with the variations in the statistics of the quantized residuals.
- E (*Estimation* lines): number of lines used to estimate the variance of unquantized prediction residuals. The higher it is, the more accurate is the prediction of rate and distortion given by the model. A serial implementation would require a very low value to keep the overhead of the rate controller low, but the proposed parallel architecture can exploit higher values, thus providing increased performance.
- L (*Leftover* lines): number of lines that are excluded from the information on the actual rate produced by the coding thread. This parameter should ideally be zero to provide information on the actual rate of the whole slice, but it is required because of the time the rate controller needs to perform the allocation of the quantization step sizes.

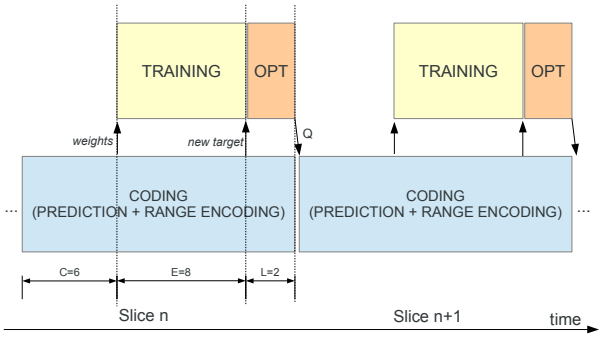

Fig. 3. Parallel implementation.

3.2. Low-memory range coding

The full compression algorithm, designed in [1] to be an extension of CCSDS-123 to lossy compression, employs an adaptive range coder [13] as entropy coding stage. This choice is motivated by the need of a block coder to replace the Golomb coder present in the lossless standard in order to achieve rates lower than 1 bpp. The range coder is closely related to arithmetic coding but it allows for faster implementation. The implementation proposed in [1] keeps four first-order models for the residuals in each band, updating them as the encoding proceeds. The use of four models follows the approach proposed in [14], where a ternary tag is first used to denote a positive, negative or zero residual, then one model is used for residuals whose magnitude is lower than a threshold and, finally, the last two models are used for the most significant and least significant bytes of residuals with magnitude greater than the threshold. This approach was essentially motivated by the fact that a prediction residual represented over 16 bits can take 2^{16} values, thus storing the frequency count for each of those symbols becomes unmanageable in terms of memory and estimating so many counts leads to a very long adaptation time. Instead, considering 1 byte at a time allows to have only 256 symbols. However, this solution still requires a significant amount of memory, typically in the order of some megabytes for hyperspectral images, because it requires to store and maintain separate statistical models for each of the bands. This is especially important for hardware implementations where one does not want to store the models on slow external memories. We address this excessive memory occupation by proposing two modifications. 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 single model is 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-

Table 1. Performance comparison.

	PARALLEL (proposed)			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

Table 2. Low-memory range coder penalty (bpp).

IMAGE	$Q = 1$	$Q = 5$	$Q = 15$	$Q = 25$
AIRS_GRAN9	0.039	0.031	0.012	0.003
AVIRIS_SC0	0.050	0.045	0.044	0.039
CRISM-SC214-NUC	0.140	0.090	0.043	0.029
VGT1_1B	0.094	0.055	0.043	0.037

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 Sec. 4 we show that the suboptimality of the new scheme is limited so that the advantages in terms of memory reduction significantly outweigh the losses.

4. EXPERIMENTAL RESULTS

We present several experimental results obtained on a subset of the hyperspectral and multispectral images belonging to the test set defined by the Multispectral and Hyperspectral Data Compression (MHDC) working group of the CCSDS. The purpose of the tests is to evaluate the performance of the proposed architecture with respect to the baseline serial algorithm, and with respect to a state of the art method for onboard compression that employs a transform coder based on the Discrete Wavelet Transform as defined in the CCSDS-122 recommendation [2], paired with a low-complexity spectral transform such as the Pairwise Orthogonal Transform (POT) [3], which is an approximation to the Kahrunen-Loève transform. The implementation is publicly available online [15]. The transform coder performs rate control by means of the reverse waterfill algorithm [9]. All the results are obtained with $C = 6$, $E = 8$, and $L = 2$ for the parallel architecture and with $E = 8$ for the serial one in order to have a fair comparison. Table 1 reports the SNR and maximum absolute difference (MAD) metrics for the parallel architecture with low-memory range coder, the serial algorithm from [1] and the transform coding method. It can be noticed that the suboptimality of the new architecture with respect to the serial version is limited to a drop in SNR smaller than 0.5 dB and typically has the same performance in terms of MAD. Hence, the proposed architecture is still competitive when compared to state-of-the-art transform coding, offering significantly higher SNR (gains ranging from 1 to 30 dB) and MAD. Table 2, instead, is focused on the penalty in terms of output rate incurred when using the low-memory version of the range coder. The values in the table show the increase in output rate, measured in bits-per-pixel (bpp), with respect to the high-memory version. To this purpose we operate the compression algorithm in near-lossless mode, *i.e.*, without rate control and with a single quantization step Q for the whole image. It can be noticed that the penalty is very limited: typically less than 0.10 bpp at lossless quality, and much less for lossy compression. We also notice that the penalty decreases with increasing value of Q (coarser quality).

5. CONCLUSIONS

This paper presented an architecture for rate-controlled predictive coding of hyperspectral and multispectral images onboard of spacecrafts. The proposed architecture builds on [1] and aims at achieving a hardware-friendly solution to onboard compression of multispectral and hyperspectral images. We showed that it is possible to devise a pipelined architecture in which the rate control algorithm works as an independent module, in parallel with the main coding module. Moreover, a more memory-efficient entropy coding stage has been devised due to the high memory demands of the statistical model of the range coder proposed in [1]. Extensive experimental results show the remarkable performance of the proposed architecture when compared to state of the art coders for onboard compression. A full FPGA implementation of the algorithm is ongoing, and its detailed hardware architecture and performance analysis will be reported in a forthcoming paper.

6. REFERENCES

[1] 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*, to appear 2014, available as preprint on IEEE Xplore.

[2] Consultative Committee for Space Data Systems (CCSDS), "Image Data Compression," *Blue Book*, November 2005. [Online]. Available: http://public.ccsds.org/publications/archive/122x0b1c3.pdf

[3] I. Blanes and J. 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] L.-S. Lan and I. S. Reed, "Fast approximate Karhunen-Loève transform with applications to digital image coding," in *Visual Communications' 93*. International Society for Optics and Photonics, 1993, pp. 444–455.

[5] A. B. Kiely and M. 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.

[6] F. Rizzo, B. Carpentieri, G. Motta, and J. 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] 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," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 4, pp. 1892–1904, 2010.

[8] B. Aiazzi, P. Alba, L. Alparone, and S. Baronti, "Lossless compression of multi/hyper-spectral imagery based on a 3-d fuzzy prediction," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 5, pp. 2287–2294, 1999.

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

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

[11] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2005.

[12] S. H. Cho and V. J. 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.

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

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

[15] Group on Interactive Coding of Images, "Delta software (a futurible CCSDS implementation)," 2013. [Online]. Available: http://gici.uab.cat/GiciWebPage/delta.php