Efficient Lossless Compression of 4D Hyperspectral Image Data

Hongda Shen, W. David Pan and Yuhang Dong
Dept. of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, AL 35899, USA
Emails:{hs0017, pand, yd0009}@uah.edu

Abstract— *Time-lapse hyperspectral imaging technology has been used for various remote sensing applications due to its excellent capability of monitoring regions of interest over a period of time. However, large data volume of four-dimensional hyperspectral imagery demands for massive data compression techniques. While conventional 3D hyperspectral data compression methods exploit only spatial and spectral correlations, we proposed a novel lossless compression algorithm that can achieve significant gains on compression efficiency by also taking into account temporal correlations inherent in the dataset. Experimental results demonstrated the effectiveness of the proposed algorithm.*

Keywords: Lossless compression, 4D hyperspectral image, temporal correlation, LMS, correntropy

1. Introduction

With more advanced remote sensing sensors being used, the spatial and spectral resolutions of the images captured by those sensors has increased rapidly, which naturally leads to large data volume. Hyperspectral imaging technology collects the image information across a wide-range electromagnetic spectrum with fine wavelength resolution. Hence, a hyperspectral image (HSI) is a three dimensional data cube with two spatial dimensions and one spectral dimension. Given the fact that most of remote sensing sensors collect data using either 12-bit or 16-bit precision, the size of a hyperspectral image cube is typically very large.

Time-lapse hyperspectral imagery is a sequence of 3D HSIs captured over the same scene but at different time stamps (often at a fixed time interval). Actually, time-lapse hyperspectral imagery can be considered as a 4D dataset whose size increases significantly with the total number of time stamps. Fig. 1 shows an illustration of one time-lapse hyperspectral image dataset. Each stack represents one 3D HSI. Furthermore, more stacks will be captured by the HSI sensor with the time. Particularly, in the extreme case of 4D HSI data streaming, the captured data volume accumulates very fast. This huge data volume does not only slow down the data transmission within the limited bandwidth condition but also requires more storage space which could be very expensive in many remote sensing applications. Data compression techniques provide a good solution to these problems. As captured images are most likely at high fidelity for the accuracy demanding applications, lossless compression is often chosen for these sensors over the lossy compression.

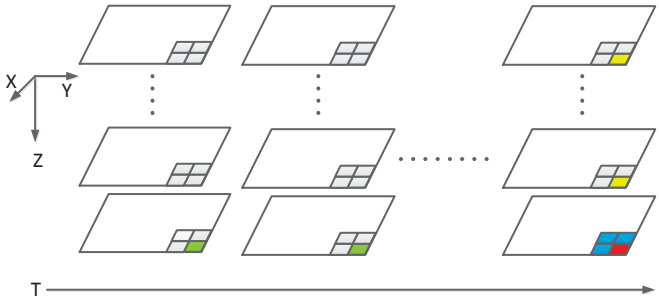


Fig. 1: A 4D time-lapse hyperspectral image dataset, where X and Y are the spatial directions, and Z is the spectral direction.

Large efforts have been made to develop a lossless compression algorithm for 3D HSI. LOCO-I [1] and 2D-CALIC [2] utilize spatial redundancy to reduce the entropy of prediction residuals. Since there exists strong spectral correlation, 3D methods including 3D-CALIC [3], M-CALIC [4], LUT [5] and its variants, SLSQ [6] and CCAP [7] take this spectral correlation into account and yield better compression performance. Also, some transform-based methods, such as SPIHT [8], SPECK [9], etc., can be easily extended to lossless compression even though they were designed for lossy compression. In addition to the goal of reducing the entropy of either prediction residuals or transform coefficients, low computational complexity is another influential factor because many sensing platforms have very limited computing resources. Therefore, a new method named as the “Fast Lossless" (FL) method, proposed by the NASA Jet Propulsion Lab (JPL) in [10], was selected as the core predictor in the CCSDS new Standard for Multispectral and Hyperspectral Data Compression [11], to deal with 3D HSI data compression.

This is a regular research paper. W.D. Pan is the contact author.

Fig. 2: Sample time-lapse hyperspectral image datasets at different time instants (from top to bottom: *Levada*, *Nogueiro* and *Gualtar*).

To give an idea on the 4D image datasets tested in this work, Fig. 2 shows the *Levada* sequence. Detailed information about the *Levada* sequence can be found in [12]. Note that only 2D color-rendered RGB images are shown in Fig. 2 instead of the actual HSI data for display purpose. Since time-lapse HSIs are captured over the same scene at different time instants with gradually changing natural illumination, there exists great similarity among these HSIs at each time instant marked in Fig. 2. Therefore, this temporal correlation can be further exploited to improve the overall compression efficiency. To the best of our knowledge, there is very few prior work on lossless compression of 4D time-lapse HSI data in the literature. [13] proposed a 4D lossless compression algorithm, albeit lacking details on the prediction algorithms used for prediction. On the other hand, in [14], a combination of Karhunen-Loeve Transform (KLT), Discrete Wavelet Transform (DWT) and JPEG 2000 has been applied to reduce the spectral and temporal redundancy of 4D remote sensing image data. However, it is a lossy compression method.

In this work, we conducted an information-theoretic analysis on the amount of compression achievable on 4D HSI based on conditional entropy, by taking into account spectral and temporal correlations. We then proposed a low-complexity correntropy-based least mean square (CLMS) learning algorithm, which was employed for the first time as a predictor to achieve higher data compression by better adapting to the underlying statistics of HSI data.

The rest of this paper is organized as follows. Section 2 introduces an information theoretic analysis framework for time-lapse HSI lossless compression. Section 3 reviews the CLMS learning algorithm. Furthermore, the proposed lossless compression engine based on CLMS learning is presented in detail. Experimental results are given in the Section 4. The paper is concluded in Section 5 with a discussion on the further work.

2. Problem Analysis

In order to evaluate the potential amount of compression we can achieve on the 4D dataset, we conducted an information-theoretic analysis. Let X_j^t be a 4D hyperspectral image source at the t^{th} time instant and j^{th} spectral band producing K different pixel values v_i $(i = 1, \cdots, K)$. Then the entropy of this source is computed based on the probabilities $p(v_i)$ of these values by

$$H(X_j^t) = -\sum_{i=1}^{K} p(v_i) \cdot log_2 \left[p(v_i)\right]. \quad (1)$$

If we assume that there are no dependencies between these pixel values for X_j^t, at least $H(X_j^t)$ bits must be spent on average for each pixel of this source. However, for the 4D hyperspectral images, this assumption does not hold given the existence of strong spectral and temporal correlations. The value of a particular pixel might depend on some other pixels from its spatial, spectral or temporal neighborhoods. Therefore, these correlations can be exploited to reduce the $H(X_j^t)$, i.e., less bits spent on average after compression. Furthermore, the conditional entropy of this time-lapse hyperspectral image source can be computed as follows:

$$H(X_j^t|C_j^t) = -\sum_{i=1}^{K} p(v_i|C_j^t) \cdot log_2 \left[p(v_i|C_j^t)\right]. \quad (2)$$

where C_j^t denoted as *context*, which represents a group of correlated pixels. As long as there is any correlation between the context C_j^t and the current pixel, $H(X_j^t|C_j^t) < H(X_j^t)$ always holds, in other words, fewer bits are required after compression.

The choice of context largely determines how much compression we can achieve by using prediction-based lossless

compression schemes. Intuitively, highly-correlated pixels are expected to be included into the context. Given the consideration that spectral and temporal correlations are typically much stronger than spatial correlation in hyperspectral images, our focus in this work is on spectral and temporal decorrelation. In fact, recent research [15], has shown that explicit spatial decorrelation is not always necessary to achieve good performance [16]. Also, in contrast to nonlinearity nature of spatial decorrelation, a linear prediction scheme is believed to be adequate for spectral and/or temporal prediction because of high degree of correlations [16]. In Section 4, we will investigate the actual compression gains using different combinations of context pixels.

3. The Algorithm

Linear prediction based lossless compression method uses a linear combination of those encoded pixels (causal context pixels) adjacent to the current pixel as its estimate. For 4D time-lapse HSI lossless compression, a linear prediction can be generalized as follows:

$$\widehat{x}^{t,j}_{m,n} = \mathbf{w}^T_{t,j}\mathbf{y}^{t,j}_{m,n}. \tag{3}$$

where $\widehat{x}^{t,j}_{m,n}$ represents an estimate of a pixel, $x^{t,j}_{m,n}$, at spatial location (m,n), j^{th} band and t^{th} time frame while $\mathbf{y}^{t,j}_{m,n}$ and $\mathbf{w}_{t,j}$ represent its causal context pixels and linear weights respectively. Note that only adjacent spectral bands and bands from previous time frames are included in this context as mentioned in Section 2.

Prediction residuals are generated by subtracting the actual pixel values from their estimates and then encoded using entropy coders such as Golomb-Rice Codes (GRC) [17] and Arithmetic Codes (AC) [18]. In order to produce accurate estimates, linear weights must be adapted to the local statistics of pixels in the time-lapse HSI data. Recently, learning algorithms have gained some success to optimize these weights in the applications of lossless compression of 3D HSI data [10], [19]. The FL method has been selected as a new standard by CCSDS for its low-complexity and effectiveness. The core learning algorithm of FL method is least mean square (LMS). Traditional LMS methods use mean square error (MSE) as the cost function. However, it is well known that MSE is the optimal cost function for Gaussian distributed signal [20], whereas the prediction residuals more likely follow a Laplacian or Geometric distribution [1]. So the performance of the conventional LMS predictor, for example, the FL method may degrade in presence of non-Gaussian signals, especially in those very structured regions of one image. Some similar observations have been noticed in [19]. To improve the robustness of the predictor, we introduce an adaptive learning based on the Maximum Correntropy Criterion (MCC) [20].

3.1 Correntropy-Based LMS (CLMS) Cost Function

Correntropy was developed as a local similarity measure between two random variables X and Y in [21], defined by:

$$V_\sigma(X,Y) = E\left[\kappa_\sigma(X-Y)\right], \tag{4}$$

where κ_σ is a positive definite kernel with kernel width controlled by the parameter σ, and the expectation $E(\cdot)$ is practically computed using sample arithmetic average. By following [20], we choose the normalized Gaussian kernel with variance σ as the kernel $\kappa_\sigma(\cdot) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\cdot)^2}{2\sigma^2}}$.

In [21], Taylor series expansion was applied on the exponential term in the kernel in Eq. (4) so that the Correntropy can be viewed as a generalized correlation function containing even higher order moments of the error signal $X-Y$. Also, it is justified in [21] that localization introduced by the kernel can reduce the detrimental effects of outliers and impulsive noise, while second-order statistics, like MSE, may suffer from bias in these conditions. The good behavior of second order moment and fast convergence of the higher order moments are combined into this Correntropy measure. An adaptive filter was developed by replacing the conventional MSE with this Correntropy as the cost function. The detailed properties of Correntropy with derivation and analysis can be found in [21]. Assume we have a pair of random variables with a finite number of samples $\{d_i, y_i\}_{i=1}^N$ where N is the number of samples in each random variable. For example, d_i and y_i can be viewed as the actual pixel value and its estimate, respectively, in this work. Furthermore, the estimate y_i can be computed as $y_i = \mathbf{W}_i^T\mathbf{X}_i$, a linear weighted average of input vector $\mathbf{X}_i$. The Correntropy based cost function, at n^{th} time instant, can be written as:

$$J_n = \frac{1}{N\sqrt{2\pi}\sigma}\sum_{i=n-N+1}^{n}\exp\left[\frac{-\left(d_i-\mathbf{W}_n^T\mathbf{X}_n\right)^2}{2\sigma^2}\right], \tag{5}$$

where $\mathbf{W}_n$ is the filter weight at n^{th} time instant. To find the weight $\mathbf{W}$ to maximize this cost function analytically, iterative gradient descent method is used with a small learning rate μ. After computing the gradient of J_n with respect to $\mathbf{W}_n$, we obtain:

$$\mathbf{W}_{n+1} = \mathbf{W}_n + \frac{\mu}{N\sqrt{2\pi}\sigma^3}\sum_{i=n-N+1}^{n}\left[\exp\left(\frac{-{e_i}^2}{2\sigma^2}\right)e_i\mathbf{X}_i\right], \tag{6}$$

where $e_i = d_i - \mathbf{W}_n^T\mathbf{X}_n$. Inspired by the stochastic gradient, N is set to 1 to approximate the sum in Eq. (6). Therefore,

$$\mathbf{W}_{n+1} = \mathbf{W}_n + \frac{\mu}{\sqrt{2\pi}\sigma^3}\exp\left(\frac{-{e_n}^2}{2\sigma^2}\right)e_n\mathbf{X}_n, \tag{7}$$

which is very similar to the weight updating function of LMS. In fact, this Correntropy-induced updating function can be viewed as LMS with a self-adjusting learning rate,

which reflects the outlier rejection property of the Correntropy. It is worth noting that this CLMS is more robust to the outliers with almost no additional cost of algorithmic complexity compared to the conventional LMS.

In many remote sensing applications, images of scenes have complex structures as shown in Fig. 2. These structures will be reflected in the hyperspectral images as strong edges and corners. This non-linearity property of the image signal directly contributes to relatively larger residual values and consequently outliers for the residual data distribution because linear predictors cannot fully reduce the redundancy in these cases. To improve the compression performance, we utilize the CLMS to tackle structured regions in the HSI data for its outlier rejection property. On the other hand, convergence speed of the adaptive filtering plays a crucial role in the prediction. Slow convergence often leads to less accurate estimation in the prediction. CLMS provides a robust performance in the non-Gaussian condition with a faster convergence compared to the conventional LMS. Therefore, CLMS can help greatly enhance the prediction accuracy, which will further contribute to better compression.

3.2 CLMS Based Predictor

To eliminate spatial correlation effect, local mean subtraction is conducted in every band of these datasets. In Fig. 1, suppose the red pixel is the one we are predicting and the arithmetic average of the three blue pixels from its spatial causal neighborhood is computed and subtracted from the red pixel value. We apply this local mean subtraction to every pixel in the dataset. Denote N_s and N_t as the number of pixels from previous spectral bands at the current time instant (yellow pixels in Fig. 1) and the number of pixels from the same spectral bands from previous time frames (green pixels in Fig. 1), respectively. Then we define the aforementioned causal context as the feature vector $\mathbf{X}$ (shown in Eq. 7) for our CLMS learning algorithm. In order to simplify the expression, we replace $x_{m,n}^{t,j}$ defined in Eq. (3) with $x^{t,j}$. The causal context is constructed as $C_j^t = \left[x^{t,j-1}, x^{t,j-2} \cdots x^{t,j-N_s}, x^{t-1,j}, x^{t-2,j} \cdots x^{t-N_t,j}\right]$. Thus, for all pixels in one specific spectral band at any time frame, the number of their context pixels are always the same according to $N_s + N_t$ determined by the user.

For each band, we initialize the $\mathbf{w}_{t,j}$ to be all zeros and carry on the CLMS learning algorithm to all the pixels in a raster-scan order when it reaches the end of this band. Once the algorithm stops, the weight vector will be initialized again for the next band until the end of the entire HSI dataset. Algorithm 1 shows the details of the whole procedure of this CLMS predictor. We emphasize that all the residuals will be mapped to integers before sending to entropy encoder in a reversible manner.

Algorithm 1 CLMS Predictor

Initialize:
1) T (# of time frames)
2) B (# of spectral bands for each time frame)
3) $\mu = 0.3$ and $\sigma = 50$
4) Local mean subtracted data X
for t = 1:T **do**
  for b = 1:B **do**
    initialize: $\mathbf{w} = \mathbf{0}$
    for each pixel in this band **do**
      Output residual $x - \widehat{x}$ using Eq. (4).
      Updating **w** using Eq. (7).
    end for
  end for
end for

3.3 Entropy Coding

Both the Golomb-Rice code (GRC) and the arithmetic code (AC) have been widely used to encode the prediction residuals in hyperspectral image compression methods. Although AC may produce slightly better coding efficiency, GRC can yield comparable performance with accurate data modelling. Also, GRC is known for its simplicity and minimal memory capacity requirement while AC very often requires much more computations. Given the limited onboard computing power in most remote sensing applications, in this work, we employ GRC on prediction residuals of time-lapse HSI data to generate the final bit sequence for the data transmission and storage. The readers are referred to [10] for details of entropy coding.

4. Experimental Results

We conducted our experiment on three 4D time-lapse HSI test datasets, *Levada*, *Gualtar* and *Nogueiro*. Basic information of these three datasets are listed in the Table 1. Detailed information of these datasets can be found in [22]. Each single HSI has the same spatial size, 1024×1344, with 33 spectral bands. Both *Gualtar* and *Nogueiro* have nine time stamps while *Levada* has seven. Note that the original data for these datasets has been mapped into $[0, 1]$ and stored using "double" data format (64 bits). So we recover the data to its original precision by applying a linear remapping. Since our algorithm is a learning-based method which predicts the value regardless of the data scale, we believe these post-processed datasets are suitable for evaluating the lossless compression algorithm. While the size of a single dataset we tested is not very large, ranging from 454.78 MB (for 7 frames) to 584.71 MB (for 9 frames), the data can easily grow to a huge size with increased number of time frames and higher spatial and spectral resolutions. Besides, HSI data streaming can become a challenging task, where efficient data compression is essential.

There are two parameters to be determined in CLMS before the prediction: σ in kernel function (shown in Eq. 7) and initial learning rate μ. Small σ value will lead to relatively large actual learning rate and vice versa. We experimented with different parameters to achieve the best results. As a result, we fix μ and σ in Eq. (7) at 0.3 and 50 in our test.

Table 1: Datasets Used.

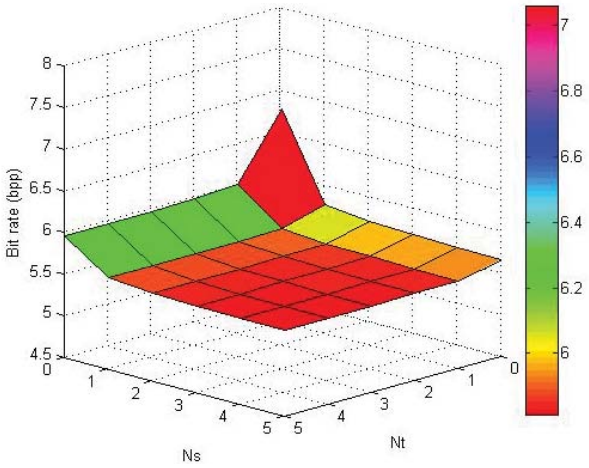
Dataset	Size	# of time frames	Precision(bits)
Levada	$1024 \times 1344 \times 33$	7	12
Gualtar	$1024 \times 1344 \times 33$	9	12
Noguerio	$1024 \times 1344 \times 33$	9	12

As discussed in Section 2, we applied our algorithm using different combinations of N_s and N_t causal pixels from spectral and temporal bands. Given the limited space, we only provide compression bit rate (bits/pixel) results for *Levada* in Table 2. As we can see, local mean subtraction without spectral and temporal decorrelation ($N_s = 0$ and $N_t = 0$), was effective in removing great deal of signal correlation as the bit rate drops from 12 to 7.0604 bits/pixel. This also indicates that it is not necessary to explicitly decorrelate spatially to achieve a competitive compression performance for time-lapse HSI data. More importantly, our algorithm can further compress 4D time-lapse HSI data by using spectral and temporal correlation. The bit rate has been reduced by approximately 1.2 bits by just adding one previous spectral band and the same spectral band from the previous time stamp in the context. Generally, the bit rate decreases, i.e., yielding less bits after compression, with more bands selected to form the learning context. Furthermore, if we fix either N_s or N_t and increase only N_t or N_s accordingly, the compression bit rate will drop as well. However, this performance improvement gradually becomes marginal as N_s or N_t increases. Fig. 3 shows three surface plots of bit rates on these test datasets, showing how the bit rate changes with different combinations of N_s and N_t.

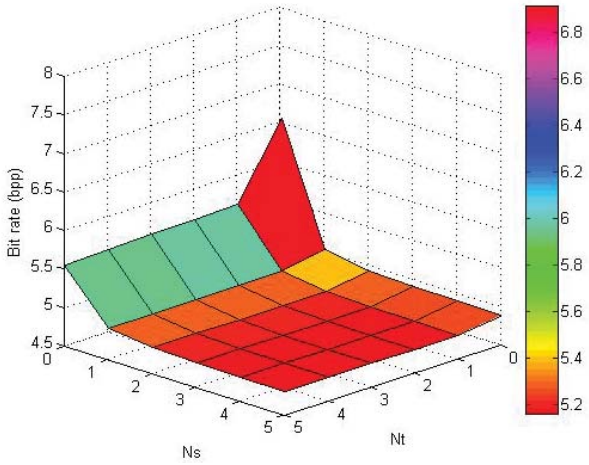
Table 2: Bit rates (bits/pixels) on "Levada".

N_s	$N_t = 0$	$N_t = 1$	$N_t = 2$	$N_t = 3$	$N_t = 4$	$N_t = 5$
0	7.0604	6.2858	6.2686	6.2431	6.2400	6.2382
1	6.0476	5.8985	5.8898	5.8813	5.8795	5.8787
2	5.9807	5.8570	5.8497	5.8414	5.8395	5.8388
3	5.9592	5.8433	5.8360	5.8279	5.8260	5.8252
4	5.9487	5.8359	5.8289	5.8211	5.8191	5.8184
5	5.9410	5.8308	5.8239	5.8162	5.8146	5.8138

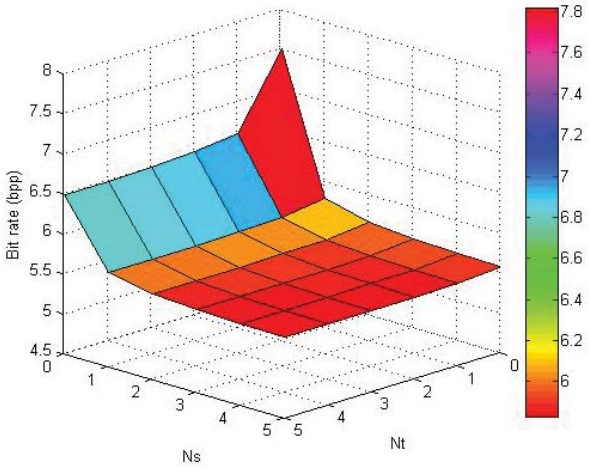
To further illustrate how bit rates respond to different combinations of N_s and N_t, we arbitrarily fix $N_s = 2$ and $N_t = 2$ separately and adjust another variable N_s or N_t from 0 to 5. The results of this experiment on three datasets have been plotted in Fig. 4. First, it is obvious that more bands used in the prediction will contribute to better prediction in terms of smaller bit rates. But this performance improvement decays really fast as what have we observed



(a) *Levada*.

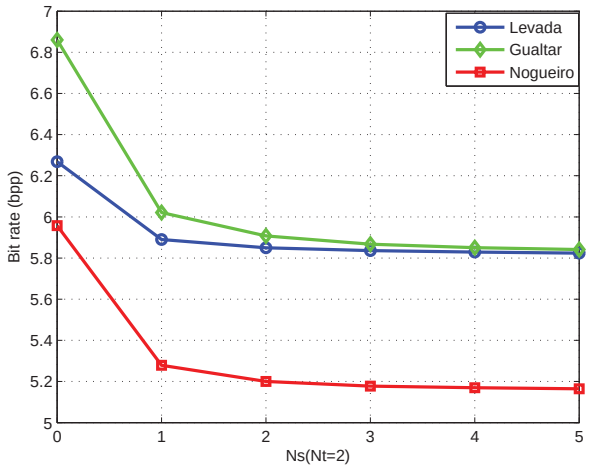


(b) *Nogueiro*.

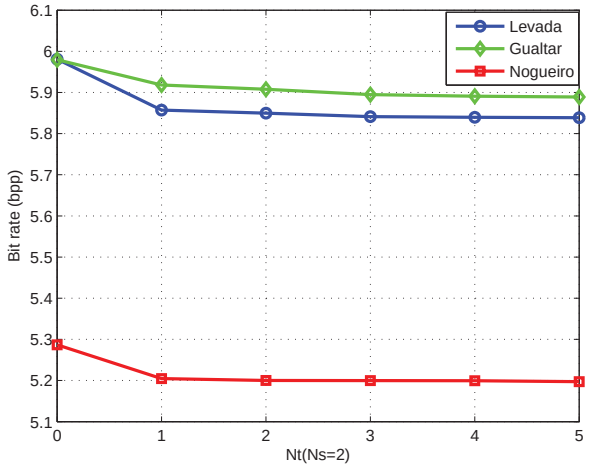


(c) *Gualtar*.

Fig. 3: Bit rate surface plots on three datasets.



(a) Bit rate vs. N_s.



(b) Bit rate vs. N_t.

Fig. 4: Bit rate changes with N_s and N_t.

in the Table 2. Moreover, we can find that the performance improvement caused by the spectral decorrelation is more noticeable than temporal decorrelation. We believe this is because for our test datasets spectral correlation is much stronger than temporal correlation especially each HSI in these 4D datasets is captured at approximately one hour interval which leads to less strong correlation temporally. If this imaging capture time interval is reduced, then adjacent HSIs will likely to share more similarities in statistics because of less illuminance condition change.

Overall, it is possible to increase N_s and N_t to achieve higher compression ratio. On the other hand, prediction using only one previous spectral band and/or the same spectral band but from last time instant will also yield good compression performance at a very low computational cost. This feature also provides great flexibility in compression performance and complexity.

5. Conclusions and Future Work

We have proposed a new predictive lossless compression algorithm for 4D time-lapse hyperspectral image data using a low-complexity Correntropy-induced LMS learning. The Correntropy based cost function seemed to be effective in capturing the non-linearity and non-Gaussian conditions of the prediction residuals of time-lapse HSI data. Experimental results have demonstrated the outstanding capability of this proposed algorithm to compress 4D time-lapse HSI data through spectral and temporal decorrelation.

Second, an information theoretic analysis based on conditional entropy has been made to provide a framework to guide and evaluate the actual compression. Increasing the number of previous bands involved in the prediction will absolutely yield better compression performance as long as they are correlated statistically with the current HSI band. We have seen the increasingly improved compression efficiency from the experimental results.

We will investigate how to fully utilize this proposed algorithm and analytic framework to handle HSI data streaming, which is more challenging but also in better need for compression. Additionally, ROI lossless compression of HSI has begun to gain attention from researchers. Recently, some work has been done to handle ROIs in HSI data. As long as ROIs can be identified accurately, we can compress the HSI data without any information loss at a high compression ratio which is comparable to lossy compression. Since our algorithm mainly utilizes spectral and temporal correlation in the prediction, it can be extended to the compression of ROIs in 4D time-lapse HSI data with minimal modifications.

References

[1] M. J. 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.

[2] X. Wu and N. Memon, "Context-based lossless interband compression-extending CALIC," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 994–1001, Jun 2000.

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

[4] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun*, vol. 45, no. 4, pp. 437–444, Apr 1997.

[5] J. Mielikainen, "Lossless compression of hyperspectral images using lookup tables," *IEEE Signal Process. Lett.*, vol. 13, no. 3, pp. 157–160, March 2006.

[6] 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.

[7] H. Wang, S. D. Babacan, and K. Sayood, "Lossless hyperspectral-image compression using context-based conditional average," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4187–4193, Dec. 2007.

[8] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun 1996.

[9] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 11, pp. 1219–1235, Nov 2004.

[10] M. Klimesh, "Low-complexity lossless compression of hyperspectral imagery via adaptive filtering," in *The Interplanetary Network Progress Report*, Jet Propulsion Laboratory, Pasadena, California, Nov. 2005, pp. 1–10.
[11] "Lossless multispectral & hyperspectral image compression CCSDS 123.0-B-1, Blue Book, May 2012," http://public.ccsds.org/publications/archive/123x0b1ec1.pdf, 2015 (accessed December 10, 2015).
[12] D. H. Foster, K. Amano, and S. M. Nascimento, "Time-lapse ratios of cone excitations in natural scenes," *Vision Research*, 2016.
[13] M. A. Mamun, X. Jia, and M. Ryan, "Sequential multispectral images compression for efficient lossless data transmission," in *2010 Second IITA Intl. Conf. on Geosci. Remote Sens.*, vol. 2, Aug 2010, pp. 615–618.
[14] J. Munoz-Gomez, J. Bartrina-Rapesta, I. Blanes, L. Jimenez-Rodriguez, F. Auli-Llinas, and J. Serra-Sagrista, "4D remote sensing image coding with JPEG2000," *Proc. SPIE*, vol. 7810, pp. 1–9, 2010.
[15] J. Mielikainen and B. Huang, "Lossless compression of hyperspectral images using clustered linear prediction with adaptive prediction length," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 6, pp. 1118–1121, Nov. 2012.
[16] E. Magli, "Multiband lossless compression of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1168–1178, April 2009.
[17] S. Golomb, "Run-length encodings (corresp.)," *IEEE Trans. Inf. Theory*, vol. 12, no. 3, pp. 399–401, Jul. 1966.
[18] A. Moffat, R. Neal, and I. H. Witten, "Arithmetic coding revisited," in *Proc. Data Compression Conf.*, Mar 1995, pp. 202–211.
[19] H. Shen, W. D. Pan, and Y. Wang, "A novel method for lossless compression of arbitrarily shaped regions of interest in hyperspectral imagery," in *Proc. 2015 IEEE SoutheastCon*, April 2015.
[20] A. Singh and J. C. Principe, "Using correntropy as a cost function in linear adaptive filters," in *Intl. Joint Conf. on Neural Netw.*, June 2009, pp. 2950–2955.
[21] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process*, vol. 55, no. 11, pp. 5286–5298, Nov 2007.
[22] "Time-lapse hyperspectral radiance images of natural scenes 2015," http://personalpages.manchester.ac.uk/staff/david.foster/Time-Lapse_HSIs/Time-Lapse_HSIs_2015.html, 2015 (accessed March 1, 2015).