On the Compression of Hyperspectral Data

Raffaele Pizzolante* and Bruno Carpentieri
Dipartimento di Informatica
Università degli Studi di Salerno
I-84084 Fisciano (SA), Italy
rpizzolante@unisa.it, bc@dia.unisa.it

Abstract

In this paper we focus on the compression of three-dimensional hyperspectral data, and review the state-of-the-art low-complexity Spectral-oriented Least SQuares (SLSQ) algorithm, which is suitable for on board implementations on airplanes or satellites. Two approaches for improving the compression performances of SLSQ are considered: band ordering and band clustering. We experimentally test the performances of SLSQ on a test data set of five NASA AVIRIS hyperspectral images and the results obtained confirm the efficiency of the algorithm.

Keywords: hyperspectral data, low complexity compression, data compression, band ordering, band clustering

1 Introduction

Data produced through hyperspectral remote sensing is increasingly used in different real-life applications, as for example in geological research or military tasks. Hyperspectral data, also called hyperspectral images, can be considered as three-dimensional data. In fact, an hyperspectral *datacube* can be intended as a sequence of bi-dimensional bands, where each band is acquired at a specific bandwidth of the electromagnetic spectrum.

Hyperspectral sensors produce daily a large amount of data. Therefore, it is essential the design of compression algorithms for the efficient transmission/storing of this kind of data. Generally, lossless compression schemes are preferred due to the high acquisition costs and to the delicate analysis that this data could undergo (for example the data could be used for target recognition or for oil fields identification). The design of compression algorithms need to take in account also the computational costs, in order to make the designed schema suitable also for "on board" implementations, by considering the limitations of the sensors in terms of computational power and memory.

Here we focus on the low complexity aspect, by considering the lossless compression Spectral-oriented Least SQuares (SLSQ) approach. SLSQ [16, 22, 20, 21] is a state-of-the-art lossless compressor that uses limited resources in terms of CPU and memory. These characteristics make SLSQ suitable for "on board" implementations. SLSQ is a predictive based schema that exploits the three-dimensional nature of the hyperspectral images.

We have investigated on how to improve the compression performances of SLSQ, by considering: a band ordering approach (see Section 4.1) and a band clustering approach (see Section 4.2).

This paper is organized as follows: Section 2 briefly describes hyperspectral data, Section 3 reviews the compression of hyperspectral data, Section 4 reviews the band ordering and the band clustering approaches. Section 5 reports the experimental results achieved on a test data set composed by five NASA AVIRIS hyperspectral images and Section 6 highlights our conclusions and future work directions.

IT CoNvergence PRActice (INPRA), volume: 1, number: 4, pp. 24-38
*Corresponding author

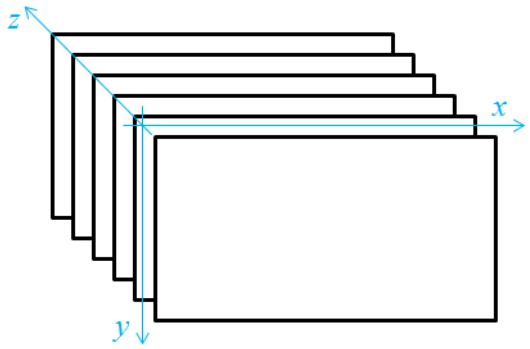


Figure 1: Three-dimensional structure of an hyperspectral image.

2 Hyperpsectal Data

An hyperspectral datacube (often referred as an hyperspectral image) can be seen as a three-dimensional data item (see Figure 1), since it is composed by a sequence of bands, where each band is a bi-dimensional collection of data. Thus, each pixel is represented by a vector of hundreds of elements.

By using the data produced by the hyperspectral remote sensors it is possible to analyze the electromagnetic spectrum of a specific observed area, including the frequencies of ultraviolet and infrared rays, which are not perceptible by the human vision system.

Generally, hyperspectral images are composed of up to about 250 spectral channels (bands). Each band has a bandwidth of about 10 nanometers (nm). As for example, the NASA Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) sensors produce hyperspectral images that are composed by 224 bands. The bands are acquired at wavelengths that range from 400 to 2500 nm and the spatial resolution of the sensor is in the order of 20x20 meters per pixel. Each pixel is acquired by using 12 bits Analog-to-Digital Converter (ADC), which will become 16 bits after the calibration and the correction phases [1].

One of the main objective of the AVIRIS hyperspectral images is the study of the Earth's surface and atmosphere of the Earth [**?**].

Figure 2.a, 2.b, 2.c and 2.c show a graphical representation of an AVIRIS hyperspectral image (respectively the 50-th, 100-th, 150-th and 200-th bands). Hyperspectral images are widely used in real-life applications: atmospheric, ecological, geological, archeological, mineralogical studies and researches, military applications, etc.. In geological applications, for example, the capabilities of the hyperspectral remote sensing can be useful to identify various types of minerals, or in the search of minerals and oil.

Another task in which the hyperspectral images are used is the identification of materials. This is possible since each material can be identified through its spectral signature. Therefore, the spectral signature is an unambiguous fingerprint that allows the identification tasks.

Some catalogues of different minerals and their spectral signatures are provided by different research organizations (like NASA, etc.) in order to permit an easier identification of the materials.

The ongoing development related to the technologies for hyperspectral remote sensing acquisition will allow to record higher resolution images and that will make possible a more sophisticated analysis of the target region. The exponential growth of the amount of the hyperspectral data produced brings the need of compression algorithms more and more.

3 Compression of Hyperspectral Data

Data produced by the today remote sensing technologies raise different challenges, including efficient transmission and storage [12]. The number of multispectral and hyperspectral sensors used is steadily increasing and each sensor produces many hundreds of gigabytes of data. Therefore, efficient compression is essential in order to reduce the transmission time and to minimize memory space for storage.

Since the sensors are generally positioned on airplanes or satellites and they have limited hardware capabilities, an algorithm for the compression of hyperspectral images should maintain a good trade-off among computational complexity, memory usage and compression performances. If computational complexity and memory usage are low, then we might have also "on-board" implementations.

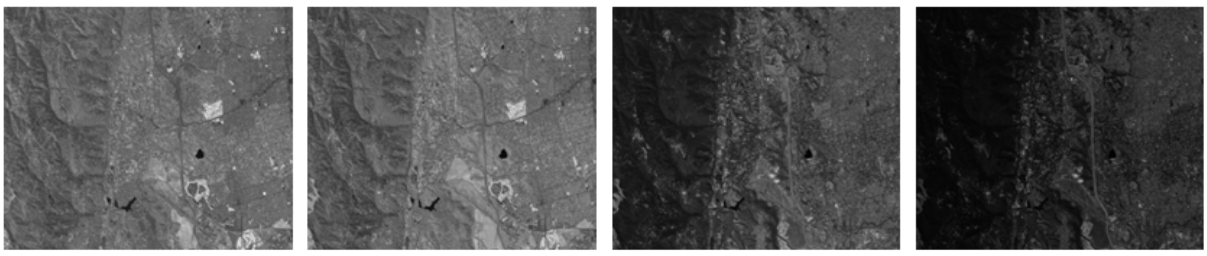
The lossless hyperspectral compression techniques that are based on predictive coding, generally, use limited resources in terms of computational power and memory. Therefore, such techniques are the preferred candidates for on-board implementations.

Some state-of-the-art lossless predictive-based techniques are Spectral-oriented Least SQuares (SLSQ) [16, 22, 20, 21], Linear Predictor (LP) [20], Fast Lossless (FL) [6], CALIC-3D [8], M-CALIC [8] and EMPORDA [23]. Other lossless techniques might have even better performances, but they are not designed for on board compression, since they have high computational or memory complexity [12].

As for example, the approaches based on the Look-Up Table (LUT) [10] raise high compression performances, by using more resources in terms of memory and CPU usage. The basic idea of LUT is to predict each pixel by using all the pixels in the current and in the previous band, by searching the nearest neighbor in the previous band, which has the same pixel value as the pixel located in the same spatial coordinates as the current pixel in the previous band.

Lossy compression techniques are generally based on 3D frequency transforms: as for examples 3-D Discrete Wavelet Transform (3D-DWT) [7], 3-D Discrete Cosine Transform (3D-DCT) [9], Karhunen-Loève transform (KLT) [15], etc.. These approaches provide high and easily scalability, but they require to maintain in memory the entire hyperspectral image at the same time.

In [12, 13, 11, 19] it is presented a scheme named Locally optimal Partitioned Vector Quantization (LPVQ) that applies Partitioned Vector Quantization (PVQ) independently to each pixel of the hyperspectral image. The variable size of the partitions are chosen adaptively and the indices are entropy coded. This technique can be used also in lossless mode, but the high resources required make it not suitable for on board implementations.



(a) 50-th (b) 100-th (c) 150-th (d) 200-th

Figure 2: Graphical representation respectively of (a) the 50-th, (b) the 100-th, (c) 150-th band and (d) the200-th band of the AVIRIS "Cuprite" image (Scene 3).

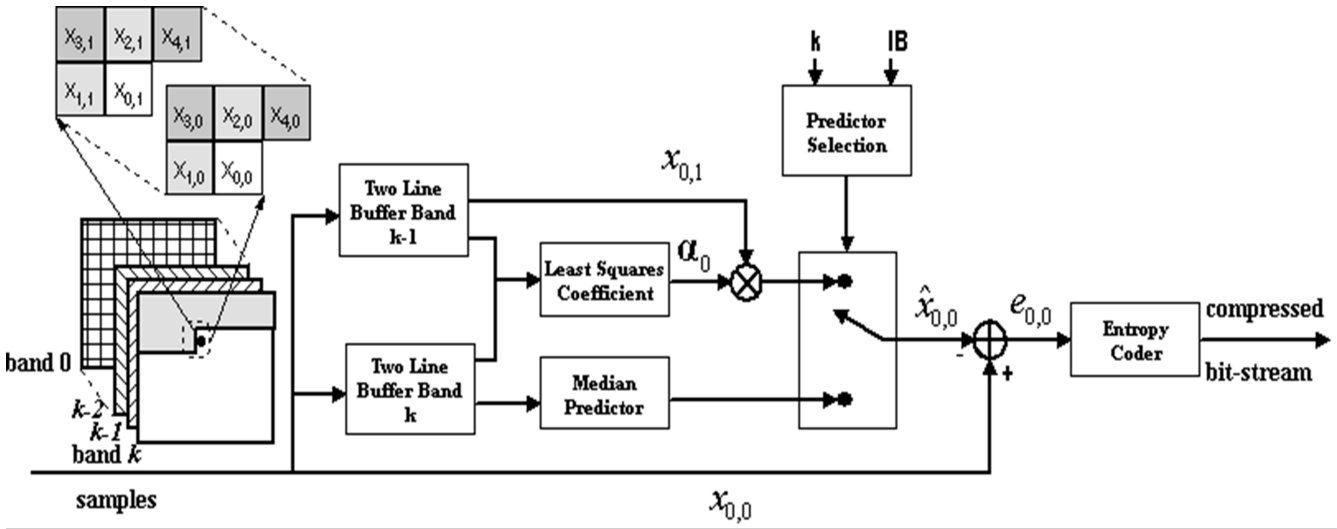


Figure 3: Block diagram of the SLSQ algorithm (from [20]).

3.1 The Spectral Oriented Least SQuares Algorithm

SLSQ (Spectral Oriented Least SQuares) is a state-of-the-art low-complexity predictive-based scheme that is suitable for on board implementations. Figure 3 shows a block diagram of the SLSQ algorithm.

A predictive-based scheme predicts each pixel, x, by using the neighboring pixels of x, which are already coded. Once the prediction is performed, the prediction error is computed as the difference between x and its prediction. Then, the prediction error is modeled and sent to an entropy encoder.

Hyperspectral data presents two types of correlation: *spatial* or *intra-band correlation* and *spectral* or *inter-band correlation*. Generally, adjacent locations are made of the same materials (spatial correlation) and a band is generally similar to the other bands (spectral correlation).

In order to exploit both spatial and spectral correlation, SLSQ uses an inter-band predictive structure, for all the bands but with the exception of a subset of bands, named Intra-Band set (IB). The bands belonging to the IB set are predicted by using a bi-dimensional predictive structure (as for example the Median Predictor [2], see Figure 3).

The prediction context for the inter-band predictive structure is obtained by considering two enumerations based on the d_{2D} and d_{3D} distance functions.

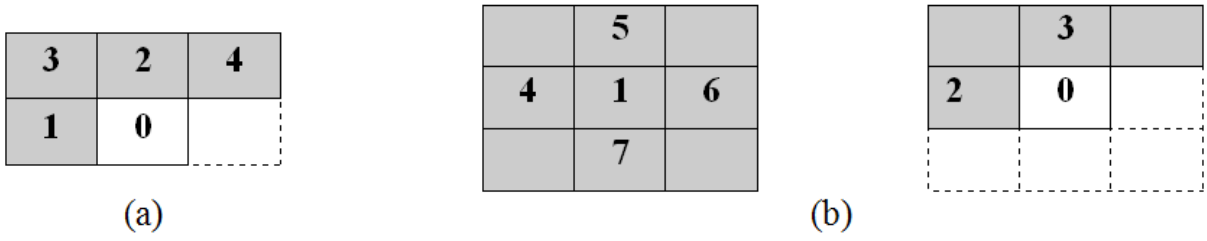


Figure 4: An example of the resulting enumeration for the intra-band prediction context (a) and for the inter-slice prediction context (b). The gray part is already coded by the algorithm.

The d_{2D} function is defined as:

$$d_{2D}(x_{n,m,k}, x_{p,q,k}) = \sqrt{(m-p)^2 + (n-q)^2} \tag{1}$$

The d_{3D} function is defined as:

$$d_{3D}(x_{n,m,k}, x_{p,q,k}) \begin{cases} \sqrt{\frac{1}{4} + (m-p)^2 + (n-q)^2}, & \text{if } i = j-1 \\ \sqrt{(m-p)^2 + (n-q)^2}, & \text{otherwise} \end{cases} \tag{2}$$

Figures 4.a and 4.b show respectively an example of the resulting intra-band prediction contexts and an example of the resulting inter-band prediction context. Let $x(i)$ denotes the i-th pixel of the intra-band context of the current pixel and let $x(i, j)$ denotes the j-th pixel in the inter-band context of $x(i)$.

According to these notations, we can denote the current pixel with $x(0,0)$. The prediction of the current sample is obtained as:

$$\hat{x}(0,0) = \sum_{j=1}^{N} \alpha_j \cdot x(0, j) \tag{3}$$

The coefficients $\alpha_0 = [\alpha_1, ..., \alpha_N]^t$ are chosen to minimize the energy of the prediction error, defined as:

$$P = \sum_{j=1}^{M} (x(i,0) - \hat{x}(i,0))^2 \tag{4}$$

By using matrix notation it is possible to rewrite P as:

$$P = (\mathbf{C}\alpha - \mathbf{X})^t \cdot (\mathbf{C}\alpha - \mathbf{X}) \tag{5}$$

where the matrices $\mathbf{C}$ and $\mathbf{X}$ are defined as:

$$\mathbf{C} = \begin{bmatrix} x(1,1) & \cdots & x(1,N) \\ \vdots & \ddots & \vdots \\ x(M,1) & \cdots & x(M,N) \end{bmatrix} \text{ and } \mathbf{X} = \begin{bmatrix} x(1,0) \\ \vdots \\ x(M,0) \end{bmatrix}. \tag{6}$$

By taking the derivate with respect to α of P and setting it to zero, the following linear system is obtained: $(\mathbf{C}^t\mathbf{C}) \cdot \alpha_0 = \mathbf{C}^t\mathbf{X}$.

The α_0 coefficients are the solutions of this linear system are and, with them, it is possible to compute the prediction. Finally, the prediction error $e = \lfloor x - \hat{x} \rceil$ is entropy coded through an arithmetic encoder.

4 Improving the Compression Performances of SLSQ

We can improve the performances of SLSQ via an online efficient band ordering (Section 4.1) or an offline band clustering (Section 4.2).

The band ordering-based approach [17] can be intended as a preprocessing stage (before the compression) for each hyperspectral image. Our preprocessing technique is not suitable for on board implementations due to the fact that is necessary to process the entire hyperspectral image and because the computational costs are too high. Instead, the computational complexity of the decompression algorithm is unaltered.

The band clustering-based approach [18] is a one-time approach that analyzes a group of hyperspectral images acquired by the same sensor. The results of our band clustering approach can be used for improve the compression performances of hyperspectral images acquired through that specific sensors. Once the one-time band clustering approach is performed, the computational complexity of both compression and decompression algorithms remains unaltered but the compression performance is enhanced.

4.1 Band Ordering

Generally, consecutive bands of an hyperspectral image have high spectral correlation, but it is possible that one or more subsets of consecutive bands might not have high correlation due to noise and other factors. By analyzing the entire hyperspectral image it is possible to observe the trend of the correlation between bands and we can exploit it in order to improve the compression performances.

An example of the band correlation trend is observable in Figure 5, which graphically reports the trends for all the scenes of the AVIRIS "Low Altitude" image. On the Y-axis it is reported the correlation value assumed by considering the band i-th (on the X-axis) and its previous band (the (i - 1)-th).

The graph of Figure 5 highlights that in the most of the cases the correlation is high among consecutive bands, but there are four subsets of consecutive bands in which the correlation is low.

In [17], we proposed a band ordering approach that uses as similarity metric the Pearson's correlation [14].

Pearson's correlation is used to determine the "similarity" between two variables, and it can assume values from -1 to 1.

Mathematically, Pearson's correlation is obtained by dividing the covariance of two random variables X and Y (σ_{xy}) and the product of the standard deviation of the random variable X (σ_x) and the random variable Y (σ_y):

$$\rho = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}$$

If the correlation assumes a positive value between the two variables, then the two variables are directly correlated and an higher correlation value indicates an higher similarity. Otherwise, if the correlation assume value equal to zero or a negative value, then the two variables are respectively not correlated or indirectly correlated.

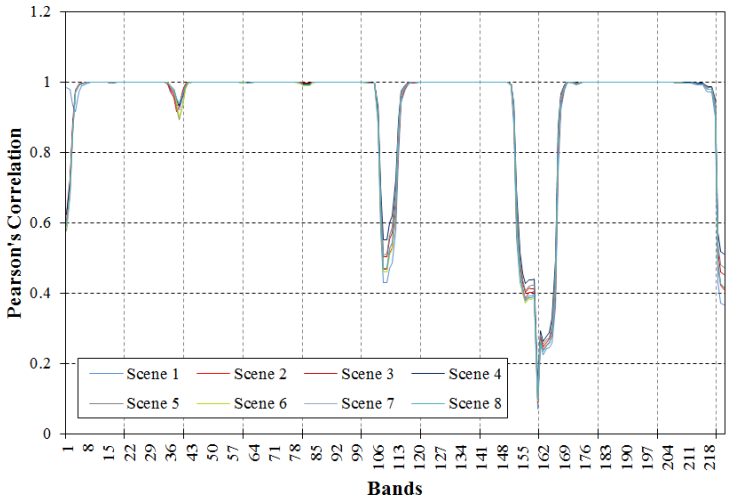


Figure 5: Trend of the band correlation for all the scenes of the AVIRIS "Low Altitude" image.

This approach can be used as preprocessing for the hyperspectral images.
Our approach can be synthetized into three main phases:

- The creation of a graph G, by using the information obtained by the hyperspectral image;
- The computation of the Minimum Spanning Tree (MST) M on the graph G;
- The computation of a slightly modified Depth First Search (DFS) visit on M.

The first phase consists into the creation of the graph $G = (V, E)$, where V is the set of the vertices and E is the set of the edges. Each band i is identified by a vertex into the graph. Each vertex is connected with any other and each edge (i, j) is weighted through the following function:

$$w(i, j) = -PearsonCorr(i, j)$$

where $PearsonCorr(i, j)$ indicates the Pearson's correlation between the band i and the band j. The minus sign is necessary due to the fact that the correlation assume values from -1 to 1 and a greater value indicates that the correlation is high, instead, the MST method considers only the minimum weights.

Once the graph G is created, it is possible to compute the MST algorithm on the graph G. By using the MST algorithm, each couple of bands (i, j) is associated with the minimum weight (that corresponds to the maximum Pearson's correlation due to the minus sign).

After this phase, it is possible to compute a slightly modified DFS that returns the final band ordering as a sequence of pairs, where each pair is composed as:

$$\langle reference\_band, band\_to\_predict \rangle$$

The reference band is necessary for each band, since more bands could have the same reference band.

4.2 Band Clustering

From the analysis of the band correlation we have observed that in each AVIRIS hyperspectral image there is a sub-set of bands that have very low correlation with respect to any other bands. We have named these bands as *NR*-bands, that can be grouped into *NR*-sets.

The basic idea behind our approach is dictated by the fact that the *NR*-bands are not efficiently predicted by a three-dimensional predictive model. Thus, our approach is based on the identification of such bands and the prediction of them through a bi-dimensional predictive model, which exploits only the spatial correlation.

Our approach is subdivided into two main stages:

- The classification of the bands and the creation of the *NR*-sets;
- The definition of the NR_F-Set, which is the final output of the approach.

Since the identification of the *NR*-bands is substantially a data clustering problem, we have used the CompLearn Toolkit (reviewed in Section 4.2.1) as data clustering tool in order to find the differences between the *NR*-bands and the other bands. CompLearn try to classify any type of files, without any preliminary background information.

4.2.1 The CompLearn Toolkit

Let us suppose that we have an homogenous group of objects. A clustering task can be viewed as a method that subdivides this group into different subsets (clusters). Each cluster contains objects that are more similar between themselves than the other objects belonging to the other clusters, with respect to a distance metric.

The distance metric can be also based on the compressibility of the data and in this case it does not include any explicit semantic knowledge, this technique is referred as clustering by compression [3, 5]. The following example synthetizes the basic concepts and the principles that are behind this model.

Let us suppose that we have an homogeneous group composed by two digital objects (files): A and B. If we compress A and B with a lossless data compressor (eg. gzip, bzip, etc.) we can indicate with L(A) and L(B) the compressed lengths (in bits) of A and B. Thus, the length of both compressed objects is L(A) + L(B). If we append the file that represents the digital object B to the file that represents the digital object A (by obtaining the concatenated file AB), then we can compress the new resulting file. The resulting length of the new compressed file shall be L(AB).

By comparing L(AB) and L(A) + L(B) is possible to determinate a "similar" metric. As for example, if $L(AB) \ll L(A) + L(B)$ this means that A and B are "similar".

Since the compression ratio indicates a great deal of important statistical information, this measure can be used as a hint in order to cluster a set of digital file we might be able to do it by considering how well they compress together in pairs.

The CompLearn toolkit, introduced in [3, 5], exploits the power of data compression and it is based on these considerations.

CompLearn, that is freely downloadable from [4], is a general-purpose method and it has been tested in a wide range of real-life applications, as for example it can classify music, language, bio sequences, etc.

Moreover, it requires no background knowledge and there are no specific parameters to configure for each domain. The result of the analysis of a set of data can is represented as an un-rooted tree, that depicts the relations among the clustered objects (represented by labeled leafs).

4.2.2 Band Clustering via CompLearn

The first stage of our approach consists into the subdivision in bands of the first scene of each of the test hyperspectral images, then we produce a file for each band.

These files are used as input for the CompLearn toolkit, which produces as output an unrooted tree. Figure 6 shows a an example of a portion of a resulting unrooted tree obtained by using CompLearn.

We can interpret the resulting tree in this mode: each band is represented by a leaf of the tree and each label identifies the band. An internal node has no label, in fact, it is used to show the relation between two or more sub-trees. Similar bands are clustered by CompLearn in the same sub-tree. Neighboring sub-trees represent band that are more related.

By analyzing the resulting trees, we have observed that the *NR*-bands are grouped in at most two sub-trees. If we define a *cut* as the elimination of a sub-tree from the clustering tree, we need at most of two cuts in order to obtain the initial *NR*-sets.

We added also the first eight bands in each initial *NR*-set, and all the bands where the prediction reference band is in the initial *NR*-set, i.e. if for the band i the compression algorithm uses as prediction reference band the band i - 1 and i - 1 belongs to the initial *NR*-set, then we consider also the band i in the final *NR*-set.

Once the *NR*-sets are computed, it is possible to compute the NR_F-Set, which is the final output of our approach.

The NR_F-Set is defined as the intersection of all the *NR*-sets, as:

$$NR_F\text{-Set} = \bigcap_{Img \in TestSet} NR\text{-set}_{Img} \tag{7}$$

where $NR\text{-set}_{Img}$ is referred to the *NR*-set for the first scene of the hyperspectral image *Img*. The bands belonging to NR_F-Set must be coded by using a bi-dimensional predictor, as for example the Median Predictor [2] instead of the three-dimensional inter-slice predictor of the SLSQ algorithm.

Figure 7 shows an example of clustering tree for the Jasper Ridge hyperspectral image, in which are identified two cuts highlighted respectively by two black circles, which are zoomed for improve the readability.

5 Experimental Results

This section reports the experimental results achieved by the SLSQ algorithm, through the band ordering preprocessing and the band clustering method.

5.1 Test Set Description

All the experiments are performed by using a well-known test set, composed by five NASA AVIRIS hyperspectral images. Each hyperspectral image is subdivided into more than one parts (denoted as scenes).

The test images are:

- Lunar Lake composed of 3 scenes;
- Moffett Field composed of 4 scenes;
- Jasper Ridge composed of 6 scenes;
- Cuprite composed of 5 scenes;
- Low Altitude composed of 8 scenes.

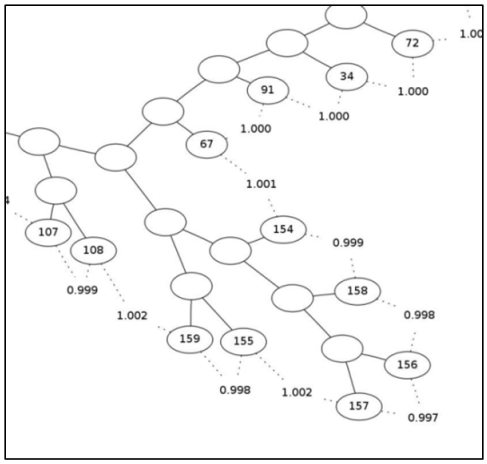


Figure 6: A portion of an example of a clustering tree obtained by using CompLearn.

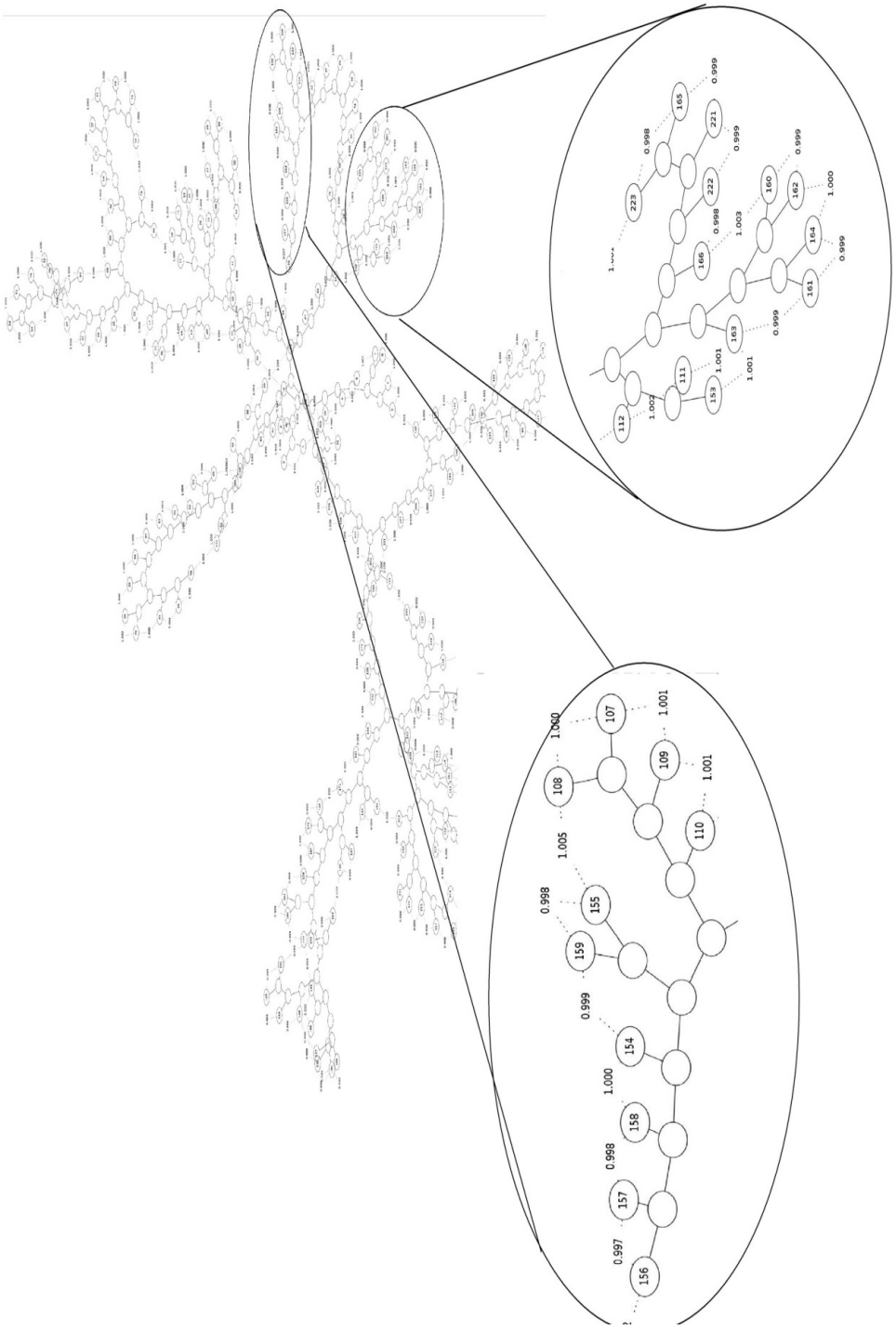


Figure 7: The two cuts on the resulting tree produced by the analysis of the first scene of Jasper Ridge (from [18]).

All the scenes of each of the test images have 614 columns, 512 lines and 224 bands, with the exceptions of the last scene of each image that has less than 512 lines. Each sample is stored by using a signed integer of 16 bits.

Table 1: Results achieved by the SLSQ algorithm.

Scenes / H. Images (C.R.)	Lunar Lake	Moffett Field	Jasper Ridge	Cuprite	Low Altitude
Scene 1	3.17	3.14	3.20	3.22	3.00
Scene 2	3.20	3.18	3.21	3.18	2.98
Scene 3	3.21	3.26	3.18	3.21	3.03
Scene 4		3.11	3.17	3.18	3.01
Scene 5			3.22	3.18	2.99
Scene 6			3.19		3.03
Scene 7					3.03
Scene 8					3.02
Average	*3.19*	*3.17*	*3.19*	*3.19*	*3.01*

Table 2: Results achieved by the SLSQ algorithm on the preprocessed hyperspectral images through our band ordering approach.

Scenes / H. Images (C.R.)	Lunar Lake	Moffett Field	Jasper Ridge	Cuprite	Low Altitude
Scene 1	3.22	3.16	3.23	3.28	3.03
Scene 2	3.25	3.20	3.24	3.22	3.01
Scene 3	3.26	3.28	3.21	3.26	3.07
Scene 4		3.14	3.20	3.23	3.05
Scene 5			3.23	3.23	3.02
Scene 6			3.23		3.06
Scene 7					3.06
Scene 8					3.05
Average	*3.24*	*3.20*	*3.22*	*3.24*	*3.04*

5.2 Results

Table 1 reports the results achieved by using the SLSQ algorithm. The columns from the second to the sixth report the achieved results in terms of *compression ratio* (C.R.), for each scene (first column) of hyperspectral images of the test set.

We have experimentally tested our band ordering approach, by preprocessing each scene of the hyperspectral images of our test set, before the compression via SLSQ. Table 2 reports the achieved results by our band ordering approach.

We have experimentally tested our band clustering approach, by performing it on the first scene of each one of the test hyperspectral image. We used the obtained NR_F-Set for the compression of all the scenes of the test hyperspectral images. Table 3 reports the results achieved by our clustering approach.

The NR_F-Set we have been obtained as reported in Table 4. Table 5 reports the number of the required cuts for each first scenes of each one of the test images.

5.3 Analysis of the Experimental Results

Table 6 reports the average results, in terms of C.R.. The first column indicates the hyperspectral images, the second column indicates the average results achieved by using the SLSQ algorithm, the third column

Table 3: Results achieved by the SLSQ algorithm by using our band clustering.

Scenes / H. Images (C.R.)	**Lunar Lake**	**Moffett Field**	**Jasper Ridge**	**Cuprite**	**Low Altitude**
Scene 1	3.19	3.16	3.21	3.24	3.00
Scene 2	3.23	3.18	3.22	3.20	2.98
Scene 3	3.24	3.26	3.19	3.23	3.03
Scene 4		3.12	3.18	3.21	3.02
Scene 5			3.22	3.20	3.00
Scene 6			3.20		3.03
Scene 7					3.03
Scene 8					3.02
Average	*3.22*	*3.18*	*3.20*	*3.22*	*3.01*

Table 4: The resulting NR_F-Set obtained by our band clustering approach.

NR_F-Set (bands)
1, 2, 3, 4, 5, 6, 7, 8, 108, 109, 110, 111, 112, 113, 154, 155,156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 222, 223, 224

Table 5: The number of the cuts required for the computation of the *NR*-set of each hyperspectral image.

Hyperspectral Image	**Number of required cuts**
Lunar Lake	2
Moffett Field	2
Jasper Ridge	2
Cuprite	2
Low Altitude	1

Table 6: Average results achieved by the SLSQ algorithm and by using our both approaches.

Scenes / H. Images (C.R.)	**SLSQ**	**Band Clustering + SLSQ**	**Band Ordering + SLSQ**
Lunar Lake	3.19	3.22	3.24
Moffett Field	3.17	3.18	3.20
Jasper Ridge	3.19	3.20	3.22
Cuprite	3.19	3.22	3.24
Low Altitude	3.01	3.01	3.04
Average	*3.15*	*3.17*	*3.19*

reports the average results achieved by using our band clustering approach and the fourth column reports the average results achieved by using our band ordering approach.

As it is possible to observe from Figure 8, which graphically reports the average achieved results of Table 6, both approaches improve the compression performances. The band ordering approach has better results than the band clustering one.

For example, in the case of Lunar Lake with the band ordering approach the results achieved, 3.24 C.R., are better than the results achieved by using the band clustering, 3.22 C.R..

However, the band clustering approach is a one-time schema, which does not alter the computational complexity of the compression and the decompression algorithms. From the other hand, the band ordering is a preprocessing and it is necessary to apply it to all the hyperspectral images, before their compression. Only for the case of Low Altitude, band clustering does not bring improvements.

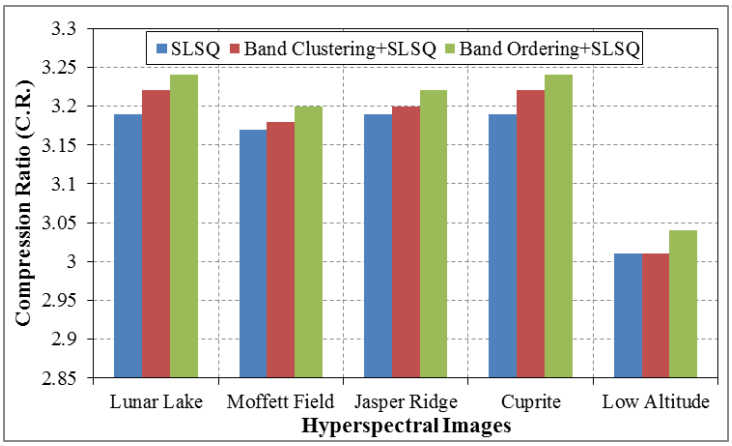


Figure 8: Graphical representation of Table 6

6 Conclusions and Future Work

In this paper, we have focused on the important issues related to the storing and the transmission of hyperspectral images, by considering the lossless compression models. We have highlighted the main goals and trade-offs that are necessary for the designing of lossless compression algorithms for hyperspectral data, which can be suitable for "on board" implementations.

We have considered SLSQ: a state-of-the-art lossless compression schema, that is suitable for on board implementations.

We have also considered two approaches for improve the compression of SLSQ. The first approach is a preprocessing schema that reorders the bands before the compression of an hyperspectral image, by considering and exploiting the correlation among bands.

The second approach is a one-time preprocessing that individuates a sub-set of bands, which are better compressed with a bi-dimensional intra-band predictor (as for example the Median Predictor) instead of the three-dimensional SLSQ predictive structure.

Future work will include additional tests of both approaches, by considering other measures for our band ordering approach and other classification tools for the band clustering approach.

Acknowledgements

We would like to thank Filippo Alfano for testing a preliminary version of our band clustering approach.

References

[1] B. Carpentieri. Hyperpectral images: Compression, visualization and band ordering. In *Proc. of the 2011 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'11), Las Vegas, USA*, pages 1023–1030, July 2011.

[2] B. Carpentieri, M. J. Weinberger, and G. Seroussi. Lossless compression of continuous-tone images. *Proceedings of the IEEE*, 88(11):1797–1809, November 2000.

[3] R. Cilibrasi. *Statistical Inference Through Data Compression*. PhD thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam, February 2007.

[4] R. Cilibrasi, A. L. Cruz, S. de Rooij, and M. Keijzer. CompLearn Website. `http://www.complearn.org/`, December 2013.

[5] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, April 2005.

[6] M. Klimesh. Low-complexity lossless compression of hyperspectral imagery via adaptive filtering. Technical Report 42-163, Jet Propulsion Laboratory, California Institute of Technology, 2005.

[7] S. Lim, K. Sohn, and C. Lee. Compression for hyperspectral images using three dimensional wavelet transform. In *Proc. of the IEEE 2001 International Geoscience and Remote Sensing Symposium (IGARSS'01), Sydney, Australia*, pages 109–111. IEEE, July 2001.

[8] E. Magli, G. Olmo, and E. Quacchio. Optimized onboard lossless and near-lossless compression of hyperspectral data using calic. *IEEE Geoscience and Remote Sensing Letters*, 1(1):21–25, January 2004.

[9] D. Markman and D. Malah. Hyperspectral image coding using 3d transforms. In *Proc. of the 8th International Conference on Image Processing (ICIP'01), Thessaloniki, Greece*, pages 114–117. IEEE, October 2001.

[10] J. Mielikainen. Optimized onboard lossless and near-lossless compression of hyperspectral data using calic. *IEEE Signal Processing Letters*, 13(3):157–160, March 2006.

[11] G. Motta, F. Rizzo, and J. A. Storer. Compression of hyperspectral imagery. In *Proc. of the Data Compression Conference (DCC'03), Snowbird, USA*, pages 317–324. IEEE, March 2003.

[12] G. Motta, F. Rizzo, and J. A. Storer. *Hyperspectral Data Compression*. Springer Science, 2006.

[13] G. Motta, F. Rizzo, J. A. Storer, and B. Carpentieri. Real-time software compression and classification of hyperspectral images. In *Proc. of Image and Signal Processing for Remote Sensing X, Maspalomas, Canary Islands, Spain*, volume 5573, pages 182–192. SPIE, September 2004.

[14] K. Pearson. Mathematical contributions to the theory of evolution.-iii. regression, heredity and panmixia. *Philosophical Transactions of The Royal Society A*, 187:253—318, January 1896.

[15] B. Penna, T. Tillo, E. Magli, and G. Olmo. Transform coding techniques for lossy hyperspectral data compression. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1408–1421, May 2007.

[16] R. Pizzolante. Lossless compression of hyperspectral imagery. In *Proc. of the 1st International Conference on Data Compression, Communication and Processing (CCP'11), Palinuro, Italy*, pages 157–162. IEEE, June 2011.

[17] R. Pizzolante and B. Carpentieri. Visualization, band ordering and compression of hyperspectral images. *Algorithm*, 5(1):76–97, February 2012.

[18] R. Pizzolante and B. Carpentieri. Band clustering for the lossless compression of aviris hyperspectral images. *ACEEE International Journal on Signal & Image Processing*, 5(1):1—14, January 2014.

[19] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer. High performance compression of hyperspectral imagery with reduced search complexity in the compressed domain. In *Proc. of the Data Compression Conference (DCC'04), Snowbird, USA*, pages 479–488. IEEE, March 2004.

[20] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer. Low-complexity lossless compression of hyperspectral imagery via linear prediction. *IEEE Signal Processing Letters*, 12(2):138–141, February 2005.

[21] F. Rizzo, B. Carpentieri, G. Motta, and J. A. Storer. Compression of hyperspectral imagery via linear prediction. In *Proc. of the 3th International Joint Conference on E-Business and Telecommunications (ICETE'06), Setúbal, Portugal*, volume 13, pages 284–291. Springer-Verlag, August 2006.

[22] F. Rizzo, G. Motta, B. Carpentieri, and J. A. Storer. Lossless compression of hyperspectral imagery: A real-time approach. In *Proc. of Image and Signal Processing for Remote Sensing X, Maspalomas, Canary Islands, Spain*, volume 5573, pages 262–272. SPIE, September 2004.

[23] J. E. Sánchez, E. Auge, J. Santaló, I. Blanes, J. Serra-Sagristá, and A. Kiely. Review and implementation of the emerging ccsds recommended standard for multispectral and hyperspectral lossless image coding. In *Proc. of the 1st International Conference on Data Compression, Communication and Processing (CCP'11), Palinuro, Italy*, pages 222–228. IEEE, June 2011.

Author Biography

Raffaele Pizzolante received his Master degree (cum laude) in Computer Science from University of Salerno (Italy) in 2011. Currently, he continues his studies as a Ph.D. student at the same university. His research interests include Data Compression, Image Processing, Digital Watermarking and Information Hiding.

Bruno Carpentieri received the "Laurea" degree in Computer Science from the University of Salerno, Salerno, Italy, and the M.A. and Ph.D. degrees in Computer Science from the Brandeis University, Waltham, MA, U.S.A. Since 1991, he has been first Assistant Professor and then Associate Professor of Computer Science at the University of Salerno (Italy). His research interests include lossless and lossy image compression, video compression and motion estimation, information hiding.