PARALLEL UNMIXING OF HYPERSPECTRAL DATA USING COMPLEXITY PURSUIT

Stefan A. Robila, Martin Butler

Department of Computer Science, Montclair State University, Montclair, NJ 07043, USA

1. INTRODUCTION

A collection of spectral images over several wavelength intervals for the same scene is called a multispectral (in case of few wide spectral bands) or hyperspectral (in case of many narrow spectral bands) image. Advances in sensor technology development have resulted in increased spectral and spatial resolution [1]. This in turn allows for significant improvements in the accuracy of the results provided through hyperspectral image processing. Fundamental to spectral imaging is the ability to correctly identify the materials by matching their lab collected spectral information (spectral signature) to the information collected in the images. Often, such direct match is not possible. Atmospheric conditions, sensor artifacts, differences in spectral resolution are only few of the factors that can affect a direct match. Furthermore, given low spatial resolution, pixels in the hyperspectral images represent mixtures of various materials [1]. Correct extraction of the basic spectra form a mixture (also called spectra unmixing) is thus of key importance, unmixing being often found at the basis of any technique that uses hyperspectral images.

Given a hyperspectral data cube, unmixing is the process of extracting a set of spectra, or endmembers and their corresponding abundances that indicate the contribution of the endmembers to each pixel vector in the cube. Both linear and non-linear mixing models have been developed, and active research continues to be performed in either direction. Of particular interest is the use of unsupervised methods for unmixing. Unlike supervised approaches (that usually start with known endmembers and then use projections, spectral mapping or least squares to extract the abundances), in unsupervised unmixing, the assumption is that no prior knowledge of the endmembers can be used. The techniques need to approximate both the endmembers and the abundances [2].

A recent ambitious approach for unsupervised unmixing is derived from Blind Source Separation (BSS) [3]. In BSS, given a multidimensional vector of observations, and assuming that such observations were produced as a linear mixture of unknown sources, the goal is to unmix the observations by regenerating the original sources and their mixing matrix [4]. From the point of view of hyperspectral images, BSS is closely aligned to the linear mixing model, with the mixing matrix being formed of the endmembers and the sources corresponding to the abundances. Additional constraints such as positivity of the results and additivity to one of the abundances need to be enforced. Various traditional algorithms such as Independent Component Analysis (ICA), Principal Component Analsysi (PCA), Nonnegative Matrix Factorization (NMF) fall under the BSS category and were proposed with various modifications as hyperspectral unmixing techniques [3,5,6]. More recently a new approach to unmixing based on concepts such as smoothness and signal complexity was proposed [7].

Complementary to unmixing, the ability to produce results in a timely fashion is critical. Hyperspectral images continue to increase in size due to both spectral (i.e. increased number of bands) and spatial (i.e. increased number of pixels in each image) resolution enhancements leading to hyperspectral cube sizes in the order of hundreds of Megabytes [8]. Moreover, most unsupervised unmixing techniques often involve iterative processes and require processing times linear or quadratic to the size of the data [9]. Applications using such data and aiming to real time or close to real time speed face thus a bottleneck on how fast a computing environment is able to process a hyperspectral image. More and more, high performance computing such as multi-core, multi-processor or distributed clusters are offered as solutions in speeding up processing. In previous research on NMF we developed a parallel algorithm for NMF based spectral unmixing and showed that, while the accuracy of the results is comparable to the sequential NMF, the speedup obtained is proportional to the number of processors used [8].

In this paper we tackle the complexity based unmixing first proposed in [7] and develop new techniques based on high performance computing that lead to significant speedup. Below we provide a brief summary of complexity based unmixing, the key bottlenecks in execution and provide elegant solutions using a parallel environment.

2. COMPLEXITY BASED UNMIXING

Complexity of a signal originates in predictability, i.e. in the ability of previous observations to predict a new observation. The level of predictability for a signal **s** is given by [7]:

$$F(\mathbf{s}) = ln\frac{\sum_{i=1}^{n}(\bar{s}-s_i)^2}{\sum_{i=1}^{n}(\tilde{s}_i-s_i)^2} \quad \text{where } \tilde{s}_i = \lambda\tilde{s}_{i-1} + (1-\lambda)s_{i-1} \tag{1}$$

$\bar{s}$ is the expected value for **s** and λ is a value between 0 and 1. The formula provides a measure of the variation between adjacent observations compared to the overall variance. A signal with high F value means that the observations do not change suddenly but rather slowly. In turn, this means that the signal is less complex.

The predictability was extended to a two dimensional space:

$$F(\mathbf{M}) = ln\frac{\sum_{i,j=1}^{r,c}\left(\bar{m}_{ij}-m_{ij}\right)^2}{\sum_{i,j=1}^{r,c}\left(\tilde{m}_{ij}-m_{ij}\right)^2} \quad \text{where } \mathbf{M}\ is\ r \times c \tag{2}$$

and $\bar{m}_{ij}$ and $\tilde{m}_{ij}$ are computed in a similar fashion as in Eq. 1 based on only the eight adjacent pixels (see Fig. 1a). In case of the expected value, the computation is done using each of these pixels equally weighted, whereas for $\tilde{m}_{ij}$ the weights are the ones given in Fig. 1b [7]

m_{i-1j-1}	m_{i-1j}	m_{i-1j+1}
m_{ij-1}	m_{ij}	m_{ij+1}
m_{i+1j-1}	m_{i+1j}	m_{i+1j+1}

(a)

0.05	0.2	0.05
0.2		0.2
0.05	0.2	0.05

(b)

Fig.1. a) Adjacent pixels contributing to estimation of m_{ij} b) weights used

An extension to three dimensional predictability (as is the case of a hyperspectral cube) is also possible. A simple approach is to add up the predictability for each band. For a cube Y formed of p bands, the predictability of the information in the cube is given by:

$$F(\mathbf{Y}) = \sum_{k=1}^{p} F(\mathbf{Y_k}) \tag{3}$$

Measures of predictability have been used for linear unmixing. In this case given the observed hyperspectral cube **R** *(r* x *c* x *l)*, we aim to find **Y** *(r* x *c* x *p)* and **W** *(l* x *p)* such that:

$$\mathbf{Y} = \mathbf{WR} \tag{4}$$

In this case, **Y** would constitute an estimate of the abundance of the endmembers and **W** would constitute the pseudoinverse of the endmembers matrix **M**:

$$\mathbf{R} \cong \mathbf{W^{-1}Y} \tag{5}$$

The Spectral and Spatial Complexity Based Unmixing proposed in ?? is starting with random initial values for **Y** and **W** and then proceeds to maximizing [7]:

$$G(\mathbf{W}, \mathbf{M}) = \alpha F(\mathbf{Y}) - \beta \sum_{k=1}^{p} ln \sum_{j=1}^{l} \left(\tilde{m}_{jk} - m_{jk}\right)^2 \tag{6}$$

where α, and β are variable parameters, Y is computed as in eq. 4, and m_{jk} refer to elements of **M**.

The algorithm was tested on various hyperspectral data cubes as well on artificial data and shown to outperform various other techniques including regular and undercomplete ICA, constrained NMF and VCA (Vertex Component Analysis).

2. PARALLEL COMPLEXITY PURSUIT

As with most of the source separation algorithms, complexity pursuit is an iterative gradient based technique heavily burdened in execution time by the size of the data. A computational complexity analysis of the gradient step derived in Eq. 6 leads to $O(p^3l^2+pl(r*c))$ complexity, i.e. an execution time for each iteration that is at least linear on the number of pixels ($r*c$) [7]. We also note that such estimate is quite optimistic. First, it based on a derivative step that uses a small neighborhood window (Fig. 1), any further changes in window size resulting in execution time increases linear on the window size. Second, the time complexity involves a constant multiplicative factor (pl) that in itself is large. Finally, as indicated, the algorithm requires multiple iterations to converge, further increasing the computational time.

To counter this we developed a modified algorithm that yields similar accuracy while leveraging a parallel computing architecture. In particular the advantage of our parallel algorithm relies on its high scalability achieved by pre-computation of common results and by distribution of the data cubes among the processing nodes. At the basis of the algorithm relies the distribution of the data to multiple processing units (see Fig. 2) as well as the use of a high level programming language environment with its synchronization modules.

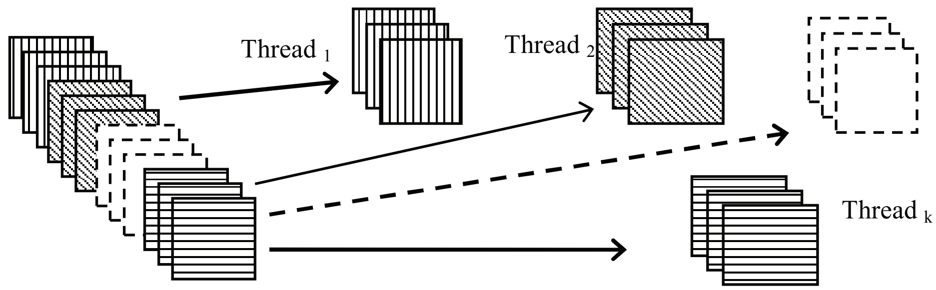

Fig. 2. Distribution of the hyperspectral data for updating by multiple execution threads. Each thread processes several of rows (i.e., an equal number of abundance bands).

Simultaneous unmixing techniques require that large amounts of data be shared among the network's processing units and frequent synchronization stages be employed in order to maintain data consistency from one algorithm iteration to another. These limitations, in turn increase the communication cost overhead often rendering cluster based implementations inefficient. Tightly coupled multiprocessor systems do not suffer from the same issues. Such systems contain two or more central processor units each with one or more computing cores that communicate directly over the system bus and share most of computer's components including the main memory. While initially parallel architectures based on shared communication bus and memory were significantly more expensive than commodity clusters, in recent years, multi-processor and multi-core architectures have become mainstream technologies in most of the off-the shelf systems. Such trend is expected only to increase in the future with predictions of many-core (leading to 1000 cores!) systems now being made.

In our research following the development of the algorithm we provide a theoretical analysis on its equivalency with the original complexity based algorithm. Furthermore we show through both computational based complexity and through experimental results that the algorithm provides a speedup in execution linear to the number of computing cores used.

11. REFERENCES

[1] C.I. Chang, *Hyperspectral imaging: Techniques for spectral detection and classification*, Springer, 2003.

[2] D. Landgrebe, "Hyperspectral image data analysis as a high dimensional signal processing problem," *IEEE Signal Processing Magazine*, vol. 19, 2002, pp. 17-28.

[3] S. Jia and Y. Qian, "A Complexity Constrained Nonnegative Matrix Factorization for Hyperspectral Unmixing," *Lecture Notes in Computer Science*, vol. 4666, 2007, p. 268.

[4] A. Plaza, P. Martinez, R. Pérez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, 2004, pp. 650-663.

[5] A. Ifarraguerri and C.I. Chang, "Unsupervised hyperspectral image analysis with projection pursuit," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, 2000, pp. 2529-2538.

[6] S. Siddiqui, S. Robila, and J. Peng, "Sparse Representations for Hyperspectral Data Classification," *IEEE International Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008*, 2008.

[7] S. Jia and Y. Qian, "Spectral and spatial complexity-based hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, 2007, pp. 3867-3879.

[8] S. Robila and L. Maciak, "Considerations on Parallelizing Nonnegative Matrix Factorization for Hyperspectral Data Unmixing," *Geoscience and Remote Sensing Letters, IEEE*, vol. 6, 2009, pp. 57-61.

[9] S. Sindhumol and M. Wilscy, "A Distributed Approach to Hyper-Spectral Image Analysis Using Support Vectors," *Intelligent Sensing and Information Processing, 2006. ICISIP 2006. Fourth International Conference on*, 2006, pp. 157-160.