PARALLEL SPARSE UNMIXING OF HYPERSPECTRAL DATA

José M. Rodríguez Alves[a], *José M. P. Nascimento*[a,b], *José M. Bioucas-Dias*[a,c], *Antonio Plaza*[d], *Vítor Silva*[e]

[a]Instituto de Telecomunicações, Lisbon, Portugal
[b]Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
[c]Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal
[d]Hyperspectral Computing Laboratory, University of Extremadura, Cáceres, Spain
[e]Instituto de Telecomunicações, DEEC, University of Coimbra, Coimbra, Portugal

ABSTRACT

In this paper, a new parallel method for sparse spectral unmixing of remotely sensed hyperspectral data on commodity graphics processing units (GPUs) is presented.

A semi-supervised approach is adopted, which relies on the increasing availability of spectral libraries of materials measured on the ground instead of resorting to endmember extraction methods. This method is based on the *spectral unmixing by splitting and augmented Lagrangian* (SUNSAL) that estimates the material's abundance fractions. The parallel method is performed in a pixel-by-pixel fashion and its implementation properly exploits the GPU architecture at low level, thus taking full advantage of the computational power of GPUs. Experimental results obtained for simulated and real hyperspectral datasets reveal significant speedup factors, up to 164 times, with regards to optimized serial implementation.

Index Terms— Hyperspectral Unmixing, Sparse Regression , Graphics Processing Unit, Parallel Methods.

1. INTRODUCTION

Remotely sensed hyperspectral images collect electromagnetic energy scattered within their ground instantaneous field of view in hundreds of nearly contiguous spectral bands with high spectral resolution [1]. This technology provides enough spectral resolution for material identification, facilitating an enormous number of applications in the fields of urban and regional planning, water resource management, environmental monitoring, food safety, counterfeit drugs detection, wild land fire tracking, oil spill and other types of chemical contamination detection, biological hazards prevention, and target detection for military and security purposes, among many others [2, 3].

Due to low spatial resolution provided by these devices and to other effects (see [1, 4] for more details), several spectrally distinct materials (also called *endmembers*) can be found within the same pixel. Thus, each pixel can be viewed as a mixture of endmember signatures.

Hyperspectral unmixing is a source separation problem which aims at estimating the number of endmembers, their spectral signatures and their abundance fractions (*i. e.*, the percentage of each endmember) [1]. Over the last decade, several algorithms have been developed to unmix remotely sensed hyperspectral data, from a geometrical and statistical point of view [1]. Some of this methods [5, 6, 7, 8] assume that dataset contains at least one pure pixel of each distinct endmember. However, when the remotely sensed data is highly mixed this assumption may not be valid [9].

A semi-supervised strategy to cope the unavailability of pure spectral signatures is to model mixed pixel observations as linear combinations of spectra from a library collected on the ground by a field spectro-radiometer [10, 11]. Thus, unmixing problem consists in findind the optimal subset of signatures in a potentially very large spectral library that can best model each mixed pixel in the scene. In practice, this is a combinatorial problem which calls for efficient linear sparse regression techniques based on sparsity-inducing regularizers, since the number of endmembers participating in a mixed pixel is usually very small compared with the dimensionality and availability of spectral libraries [10]. Some examples of sparse techniques applied to hyperspectral unmixing can be found in [12, 13, 14, 15, 16].

However, due to the high dimensionality of the hyperspectral scene and to the potentially very large library, the aforementioned techniques may be computationally very expensive, which prevents their use in time-critical applications [17, 18]. Despite the growing interest in parallel hyperspectral imaging methods and in their GPU implementations [19], no parallel implementations of sparse unmixing techniques are available on the open literature so far. This paper proposes a parallel method designed for graphics processing units (GPUs), to solve the constrained sparse regression problem. This method is based on the *spectral unmixing by splitting and augmented Lagrangian* (SUNSAL) [20] that

This work was supported by FCT-IT under project PEst-OE/EEI/LA0008/2013.

estimates the abundance fractions using the alternating direction method of multipliers (ADMM) [21]. The technique decomposes a difficult problem into a sequence of simpler ones, alleviating considerably the computational burden, and allowing a significant increase in processing speed. Additionally, this method is performed in a parallel pixel-by-pixel fashion and implemented on a GPU exploiting its architecture at low level, thus taking full advantage of its computational power, extremely high floating-point processing performance, and huge memory bandwidth.

The remainder of the paper is organized as follows. Section 2 formulates the problem and presents the fundamentals of the sparse regression problem. Section 3 describes the proposed parallel method designed for GPU. Section 4 evaluates the acceleration of the proposed method from the computational point of view. Finally, section 5 concludes the paper with some remarks.

2. SPARSE UNMIXING FORMULATION

Linear mixing model considers that a l-dimensional mixed pixel $\mathbf{y}$, of an hyperspectral image with l spectral bands, is a linear combination of endmember signatures weighted by the correspondent abundance fractions.

Let $\mathbf{A} \equiv [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$ denote a spectral library with m spectral signatures, each with l spectral bands, where it is assumed that the number of spectral signatures is much larger than the number of bands ($m \gg l$). Assuming that matrix $\mathbf{Y} \equiv [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{l \times n}$ holds n observed spectral vectors and is given by

$$\mathbf{Y} \quad = \quad \mathbf{AS} + \mathbf{N}, \tag{1}$$

where $\mathbf{S} \equiv [\mathbf{s}_1, \ldots, \mathbf{s}_n] \in \mathbb{R}^{m \times n}$ is the abundance fraction matrix and $\mathbf{N} \equiv [\mathbf{n}_1, \ldots, \mathbf{n}_n] \in \mathbb{R}^{l \times n}$ is the additive noise. To be physically meaningful [4], abundance fractions are subject to *nonnegativity constraint* (ANC) and *sum-to-one constraint* (ASC). Thus, the abundance fractions estimation problem can formulated as follows:

$$\begin{aligned} \min_{\mathbf{S}} \quad & \frac{1}{2}\|\mathbf{Y} - \mathbf{MS}\|_F^2 \\ \text{subject to :} \quad & \mathbf{S} \geq 0\,,\ \mathbf{1}_m^T\mathbf{S} = \mathbf{1}_n^T, \end{aligned} \tag{2}$$

where notation $\|(\cdot)\|_F$ stands for Frobenius norm, the inequality $\mathbf{S} \geq 0$ is to be understood componentwise, $\mathbf{1}_m$, and $\mathbf{1}_n$ denote a $1 \times m$ and $1 \times n$ column vectors filled of 1's, respectively. Due to the fact that only a few signatures contained in $\mathbf{A}$ are likely to contribute to the observed spectra $\mathbf{Y}$, $\mathbf{S}$ contains many zero values, which means that it is sparse.

2.1. SUNSAL method

The optimization problem (2) can be posed in the framework of convex optimization, where the convex term promotes the sparsity of $\mathbf{S}$. This problem is a particular case of the constrained $\ell_2 - \ell_1$ problems solved by SUNSAL, corresponding to the absence of the ℓ_1 term. SUNSAL algorithm uses the alternating direction method of multipliers (ADMM) [21] to solve (2). The pseudo-code is presented in **Algorithm 1** .

Algorithm 1 : SUNSAL

```
1:  choose $\mu > 0$, $\mathbf{U}_0$, and $\mathbf{D}_0$.
2:  $\mathbf{H} := \mathbf{A}^T\mathbf{Y}$
3:  $\mathbf{B} := \mathbf{A}^T\mathbf{A} + \mu\mathbf{I}$
4:  $\mathbf{c} := \mathbf{B}^{-1}\mathbf{1}_m\left(\mathbf{1}_m^T\mathbf{B}^{-1}\mathbf{1}_m\right)^{-1}$
5:  $\mathbf{G} := \mathbf{B}^{-1} - \mathbf{c}\mathbf{1}_m^T\mathbf{B}^{-1}$
6:  $k := 0$
7:  repeat
8:      $\mathbf{R} := \mathbf{H} + \mu\left(\mathbf{U}_k + \mathbf{D}_k\right)$
9:      $\mathbf{S}_{k+1} := \mathbf{GR} + \mathbf{c}\mathbf{1}_n^T$
10:     $\mathbf{V}_k := \mathbf{S}_{k+1} - \mathbf{D}_k$
11:     $\mathbf{U}_{k+1} := \max\{0, \mathbf{V}_k\}$
12:     $\mathbf{D}_{k+1} := \mathbf{D}_k - (\mathbf{S}_{k+1} - \mathbf{U}_{k+1})$
13:     $k := k + 1$
14: until stopping criterion is satisfied
```

3. SUNSAL IMPLEMENTATION ON GPU

SUNSAL it is highly parallelizable, since all calculations can be performed in a pixel-by-pixel basis. The first step is to map the hyperspectral image and the spectral library into GPU global memory. Fig. 1 shows the schematic of the proposed implementation for the multiplication kernel used in line 2 and line 9 of Algorithm 1. The grid contains $\lceil m/32 \rceil \times \lceil n/32 \rceil$ blocks of 32×32 and the number of threads for each block is 1024, thus all pixels in the hyperspectral image are processed in parallel. The algorithm is constructed by a chain of kernels. The first kernel computes the matrix $\mathbf{H}$ (line 2 of Algorithm 1). In order to minimize the number of global memory accesses, matrices $\mathbf{A}$ and $\mathbf{Y}$ are partitioned into subblocks of 32×32, which is the size of the block, and transferred, step by step, to the shared memory. The result for each element of $\mathbf{H}$ is the sum of several partial products. Inside the loop, another kernel compute matrix $\mathbf{R}$ (see line 8 of Algorithm 1). This kernel launches as many threads as elements that are present in $\mathbf{R}$, where each thread computes an element of the $\mathbf{H} + \mu\left(\mathbf{U}_k + \mathbf{D}_k\right)$ and stores the result in the global memory. The kernel that computes the abundance estimates, $\mathbf{S}$, on each iteration (line 9 of Algorithm 1), is decomposed on two operations: first it computes the product of matrices $\mathbf{G}$ and $\mathbf{R}$, followed by the addition of matrix $\mathbf{c}\mathbf{1}_N^T$. The scheme herein used is similar to the first kernel used to compute matrix $\mathbf{H}$ whereas the kernels to update $\mathbf{V}$ and $\mathbf{D}$ follow the same rule has the kernel used to compute $\mathbf{R}$. Finally, $\mathbf{U}$ is updated by analyzing if each element of $\mathbf{V}$ is negative.

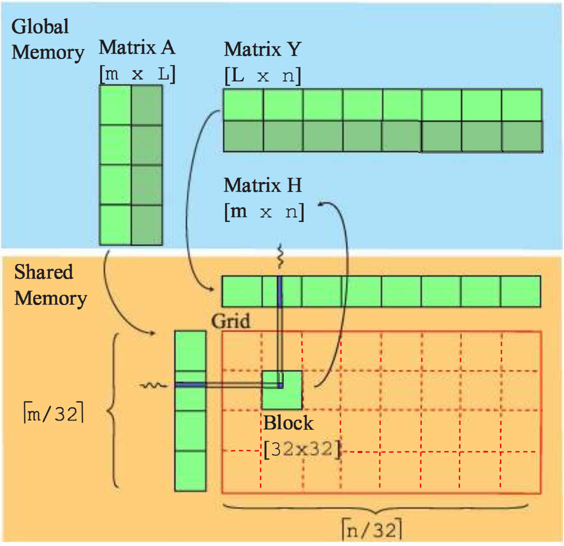

Fig. 1. Illustration of parallel SUNSAL in the GPU: multiplication kernel.

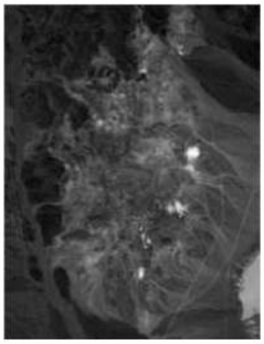

Fig. 2. Band 30 (wavelength $\lambda = 647.7nm$) of the subimage of AVIRIS Cuprite Nevada dataset.

4. PERFORMANCE EVALUATION

In this section, the sequential and parallel sparse unmixing metods are tested on a subset of the AVIRIS Cuprite Nevada dataset, available online in reflectance units. The Cuprite site has been extensively used for remote sensing experiments over the past years, is well understood mineralogically, and has several exposed minerals of interest, all included in the USGS library considered in experiments.

The subset comprises a portion of 250 × 190 pixels with 224 spectral bands between 0.4 and 2.5 μm, with a spectral resolution of 10 nm. Prior to the analysis, bands 1-2, 105-115, 150-170, and 223-224 were removed due to water absorption and low SNR in those bands, leaving a total of 188 spectral bands. Fig. 2 shows band 30 (wavelength $\lambda = 647.7nm$) of the subset considered.

Table 1. Processing times (10^3 seconds) and speedup for the Cuprite dataset ($l = 188$) and for the USGS library ($m = 342$).

	$n = 1000$	$n = 10000$	$n = 47750$
Sequential (CPU)	0.845	19.333	76.323
Parallel (GTX680)	0.036	0.124	0.465
Speedup	23.4	155.9	164.1

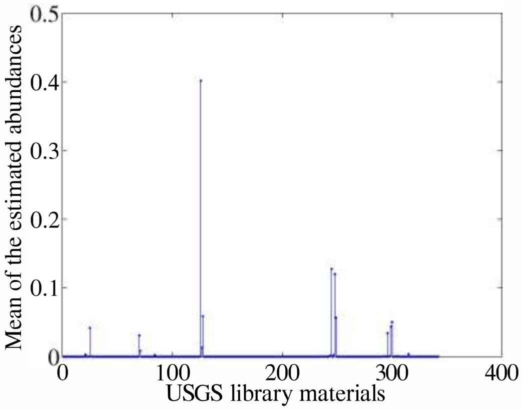

Fig. 3. Mean of the estimated abundances of all Cuprite dataset pixels as a function of the USGS library materials.

In order to compare the parallel and the sequential versions in terms of processing times, the sequential version of the method was implemented in C programming language running on one core of the Intel i7-2600 CPU, with 16 Gbyte memory and the parallel version was implemented in OpenCL programming language running on a GPU card equipped with a GTX-680 from Nvidia.

Table 1 illustrates the processing time and the speedup for the Cuprite dataset, for each column a different number of pixels is selected. The achieved speedup is approximately 164 with regards to the sequential version, which is quite remarkable taking into account that the sequential version has been carefully optimized. Note that, as expected, the acceleration factors are higher as the number of pixels to be processed is larger. The GPU total time presented in table 1 accounts for both the processing time and the time spent on host/device memory transfers, being the last one responsible for the major part.

Fig. 3, shows the mean of the estimated abundances for each material of the library. It is worse nothing that most of the materials do not contribute to the mixture, thus, the matrix is sparse. Fig. 4, presents the abundance maps of four different materials which dominate the considered scene. The results herein presented are in accordance with the USGS abundance maps of the region.

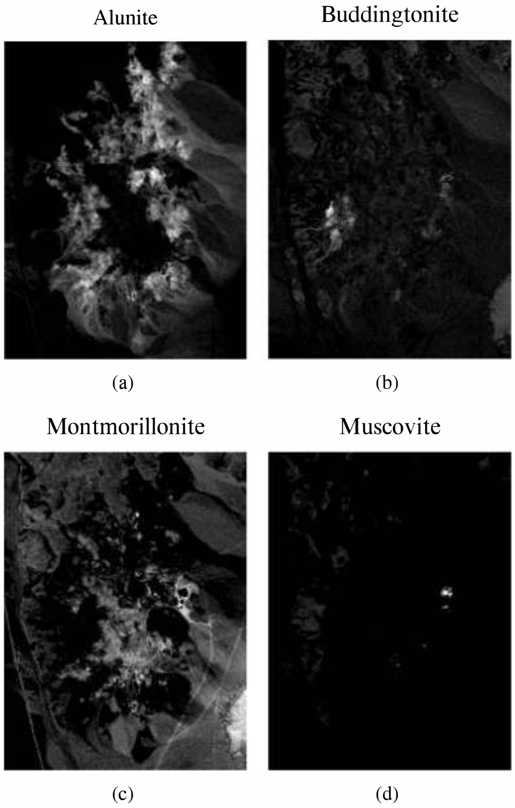

Fig. 4. Estimated abundance maps for: (a) Alunite, (b) Buddingtonite, (c) Montmorillonite, and (d) Muscovite.

5. CONCLUSIONS

In this paper a parallel method for sparse hyperspectral unmixing is proposed. The method developed on GPU has a significant speedup when compared with regards to the sequential version. Experimental results, conducted using real hyperspectral datasets collected by the AVIRIS instrument and spectral libraries publicly available from U.S. Geological Survey (USGS), indicate the potential of parallel sparse unmixing method on high performance computing environments, whenever pure pixels are not available in the dataset.

The method was designed on the multi-platform OpenCL programming language. As future work we intend to develop and test it on different hardware platforms such as FPGA board systems.

6. REFERENCES

[1] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Sel. Topics in Applied Earth Observ. and Rem. Sens.*, vol. 99, no. 1-16, 2012.

[2] M. Parente and A. Plaza, "Survey of geometric and statistical unmixing algorithms for hyperspectral images," *Proc. of the 2nd IEEE Workshop on Hyperspectral Image and Sig. Proc.: Evolution in Rem. Sens. (WHISPERS)*, pp. 1–4, Reykjavik, Iceland, Jun. 14-16, 2010.

[3] J. M. Bioucas-Dias and A. Plaza, "Hyperspectral unmixing: geometrical, statistical, and sparse regression-based approaches," *Proc. of SPIE, Image and Sig. Proc. for Rem. Sens. XVI*, vol. 7830, pp. 1–15, Toulouse, France, Sept. 20-23, 2010.

[4] J. M. P. Nascimento and J. M. Bioucas-Dias, "Does Independent Component Analysis Play a Role in Unmixing Hyperspectral Data?," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 1, pp. 175–187, 2005.

[5] J. M. P. Nascimento and J. M. Bioucas-Dias, "Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 4, pp. 898–910, 2005.

[6] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. Geosci. Rem. Sens.*, vol. 40, no. 9, pp. 2025–2041, 2002.

[7] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 3, pp. 650–663, 2004.

[8] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. Geosci. Rem. Sens.*, vol. 49, no. 11, pp. 4177 –4193, 2011.

[9] J. M. P. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of dirichlet components," *IEEE Trans. on Geosci. Rem. Sens.*, vol. 50, no. 3, pp. 863–878, 2012.

[10] M. D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. on Geosci. Rem. Sens.*, vol. 49, no. 6, pp. 2014–2039, 2011.

[11] D. M. Rogge, B. Rivard, J. Zhang, and J Feng, "Iterative spectral unmixing for optimizing per-pixel endmember sets," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 12, pp. 3725–3736, 2006.

[12] K.E. Themelis, A.A. Rontogiannis, and K.D. Koutroumbas, "A novel hierarchical bayesian approach for sparse semisupervised hyperspectral unmixing," *IEEE Trans. Sig. Proc.*, vol. 60, no. 2, pp. 585–599, 2012.

[13] X.-L. Zhao, F. Wang, T.-Z. Huang, M. K. Ng, and R. J. Plemmons, "Deblurring and sparse unmixing for hyperspectral images," *IEEE Trans. on Geosci. Rem. Sens.*, 2013, to appear.

[14] A.S. Charles, B.A. Olshausen, and C.J. Rozell, "Learning sparse codes for hyperspectral imagery," *IEEE Journal Selec. Topics in Sig. Proc.*, vol. 5, no. 5, pp. 963–978, 2011.

[15] J. B. Greer, "Sparse demixing of hyperspectral images," *IEEE Trans. Sig. Proc.*, vol. 21, no. 1, pp. 213–228, 2012.

[16] Z. Guo, T. Wittman, and S. Osher, "L1 unmixing and its application to hyperspectral image enhancement," in *Proc. SPIE 7334, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, 2009, pp. 73341M–73341M–9.

[17] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel hyperspectral image and signal processing," *IEEE Sig. Proc. Mag.*, vol. 28, no. 3, pp. 119–126, 2011.

[18] S. Sanchez, A. Paz, G. Martin, and A. Plaza, "Parallel unmixing of remotely sensed hyperspectral images on commodity graphics processing units," *Concurrency and Computation: Practice & Experience*, vol. 23, no. 13, pp. 1538–1557, 2011.

[19] A. Plaza and C.-I Chang, *High Performance Computing in Remote Sensing*, Taylor & Francis: Boca Raton, FL, 2007.

[20] J.M. Bioucas-Dias and M.A.T. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *Hyperspectral Image and Sig. Proc.: Evolution in Rem. Sens. (WHISPERS), 2010 2nd Workshop on*, 2010, pp. 1 –4.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers.," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.