GPU Implementation of Spatial–Spectral Preprocessing for Hyperspectral Unmixing

Luis Ignacio Jiménez, Gabriel Martín, Sergio Sánchez, Carlos García, Sergio Bernabé, Javier Plaza, *Senior Member, IEEE*, and Antonio Plaza, *Fellow, IEEE*

***Abstract*—Spectral unmixing pursues the identification of spectrally pure constituents, called *endmembers*, and their corresponding *abundances* in each pixel of a hyperspectral image. Most unmixing techniques have focused on the exploitation of spectral information alone. Recently, some techniques have been developed to take advantage of the complementary information provided by the spatial correlation of the pixels in the image. Computational complexity represents a major problem in these spatial–spectral techniques, as hyperspectral images contain very rich information in both the spatial and spectral domains. In this letter, we develop a computationally efficient implementation of a spatial–spectral processing algorithm that has been successfully applied prior to the spectral unmixing of the hyperspectral data. Our implementation has been optimized for the commodity graphics processing units (GPUs) and is evaluated (using both synthetic and real data) using different GPU architectures. Significant speedups can be achieved when processing hyperspectral images of different sizes. This allows for the inclusion of the proposed parallel preprocessing module in a full hyperspectral unmixing chain able to operate in real time.**

***Index Terms*—Graphics processing units (GPUs), hyperspectral unmixing, spatial–spectral preprocessing (SSPP).**

I. Introduction

THE wealth of spectral information provided by imaging spectrometers has promoted the application of hyperspectral imaging techniques in many different areas of interest [1]. In hyperspectral unmixing, endmember extraction is the process of collecting pure signature spectra of the materials present in a remotely sensed hyperspectral scene. These pure signatures are then used to decompose the scene into a set of so-called abundance fractions, representing the coverage of each endmember in each image pixel.

Several algorithms have been developed for automatic or semiautomatic identification of endmembers over the last decade [2]. A majority of the algorithms have been developed under the pure pixel assumption, i.e., they assume that the remotely sensed data contain one pure observation for each different material in the scene [3]. These algorithms often rely exclusively on the exploitation of spectral information in order to select the final set of endmembers. However, spatial information can greatly assist in the unmixing task by considering local structures latent in the data [4].

In order to also include the spatial information, several techniques have been proposed in the literature, such as the automatic morphological endmember extraction [5] or the spatial–spectral endmember extraction [6]. Furthermore, several spatial preprocessing algorithms have been developed that can be applied prior to any spectral-based endmember extraction technique. Techniques include the spatial preprocessing (SPP) [7], region-based SPP [8], and spatial–spectral preprocessing (SSPP) [9]. The goal of these preprocessing methods is to guide the search for endmembers using not only spectral but also spatial information, which can greatly assist in the selection of more spatially representative endmembers without the need to modify the endmember identification algorithm (the preprocessing can be applied as an optional step). Such SPP adds some extra computational cost to the full spectral unmixing chain. As a result, the development of efficient implementations for SPP techniques has become an important goal.

In this letter, we present a new parallel implementation of the SSPP algorithm, which has been shown to be one of the most successful SPP techniques available in the literature [9]. Our implementation has been developed for commodity graphics processing units (GPUs) [10] and tested on several GPU architectures. Synthetic scenes are used to validate the efficacy of the implementation, whereas real hyperspectral data are used to evaluate a full unmixing chain that includes our efficient preprocessing module. The results indicate that significant speedups can be achieved, allowing us to embed the SSPP into a full unmixing chain that performs in real time after including the SPP module.

The remainder of this letter is organized as follows. Section II enumerates and describes the different steps of the SSPP method. Section III describes the proposed parallel implementation for GPUs. Section IV describes the experiments conducted using synthetic data, as well as the real data experiments intended to evaluate the acceleration achieved by our parallel implementation in the context of a full hyperspectral unmixing chain. Section V concludes this letter with some remarks and hints at plausible future research lines.

II. SSPP

This section briefly outlines the SSPP algorithm in [9]. As shown in the flowchart given in Fig. 1, the SSPP method consists of the following steps.

1) *Multiscale Gaussian filtering*. This step takes as input the original hyperspectral image $\mathbf{Y}^{I\times J\times B}$, where I is the

Manuscript received February 23, 2016; revised June 17, 2016; accepted August 8, 2016. Date of publication September 8, 2016; date of current version October 12, 2016. This work was supported by Junta de Extremadura (decreto 297/2014, ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR15005).

L. I. Jiménez, S. Sánchez, J. Plaza, and A. Plaza are with the Hyperspectral Computing Laboratory, University of Extremadura, 10071 Cáceres, Spain (e-mail: luijimenez@unex.es; sersanmar@unex.es; jplaza@unex.es; aplaza@unex.es).

G. Martín is with Instituto de Telecomunicações, 1049-001 Lisbon, Portugal (e-mail: gabriel.hernandez@lx.it.pt).

C. García and S. Bernabé are with Complutense University of Madrid, 28040 Madrid, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2016.2602227

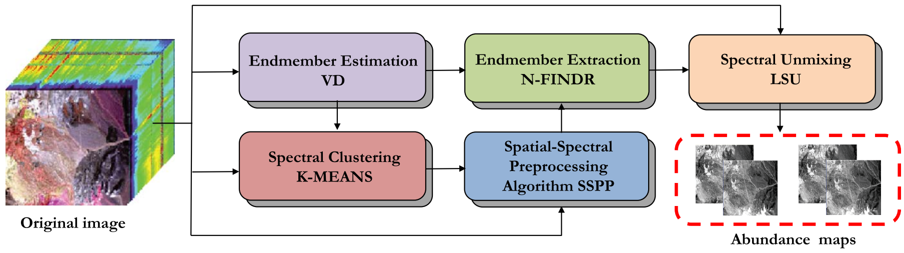

Fig. 5. Fully parallel hyperspectral unmixing chain.

MultiCPU versions as the number of endmember increases. In the GPU case, this is a consequence of the implementation strategy considered for `maxminbands` and `weights` kernels. Last but not least, the linear trend of the results suggests that real data sets with larger sizes may allow for the inclusion of SSPP in a full unmixing chain able to operate in real time.

B. Real Data Experiments

We have used two real hyperspectral data sets. The first one was collected by the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) instrument, which is operated by the NASA's Jet Propulsion Laboratory, over the Cuprite mining district in Nevada in the summer of 1997 (these data are available online).[5] The portion used in the experiments corresponds to a 350 $\times$ 350 pixel subset with 188 spectral bands in the range from 400 to 2500 μm, and a total size of 50 MB (several bands were removed due to water absorption and low signal to noise ratio in those bands). The second hyperspectral scene was also collected by AVIRIS over the World Trade Center (WTC) area in New York City on September 16, 2001 just five days after the terrorist attacks that collapsed the two main towers and other buildings in the WTC complex.[6] The full data set selected for experiments consists of 614 $\times$ 512 pixels, 224 spectral bands, and a total size of $\approx$140 MB.

The unmixing chain used in these experiments is composed of five different parallel steps (see Fig. 5). The GPU implementation of the different parts of the chain is described in [14]. The estimated number of endmembers is $p = 25$ for the Cuprite scene and $p = 31$ for the WTC scene. Table II shows the time for each step of the parallel unmixing chain (with our GPU version of SSPP embedded) and the obtained speedups in three different GPU architectures. Architectures 1 and 2 (based on NVidia GTX devices) obtain better performance than Architecture 3 (based on NVidia TESLA GPU devices). This is because the NVidia TESLA includes error checking and correction that guarantees more stable results at the expense of a slightly reduced performance. Fig. 4 shows the execution times of each step of the unmixing chain for the WTC data set. The time taken by data transfers between the CPU and the GPU is included in the execution times reported in Fig. 4. Such overhead represents 16.63%, 21.94%, and 9.4% of the total execution time for Architectures 1, 2, and 3, respectively. In the case of AVIRIS (a pushbroom instrument), the cross-track line scan time is quite fast (8.3 ms to collect 512 full pixel vectors). For real-time performance, the WTC image (512 $\times$ 614 pixel vectors) needs to be processed in approximately 5.2 s, which results from a data collection rate of approximately 27 MB/s. As shown in Table II, the execution of the SSPP algorithm is always below this threshold. In addition, the full unmixing chain is below the threshold in the case of Architecture 2.

[5]http://aviris.jpl.nasa.gov
[6]http://speclab.cr.usgs.gov/wtc/

V. Conclusions and Future Lines

In this letter, we have developed a new GPU implementation of the SSPP algorithm. The obtained results indicate that it is possible to achieve significant speedups by overlapping the execution of the algorithm in the CPU/GPU. The parallel SSPP has been embedded into a full real-time unmixing chain. Future work will focus on improving this implementation by studying different clustering and endmember extraction algorithms.

References

[1] A. Goetz, G. Vane, J. Solomon, and B. Rock, "Imaging spectrometry for earth remote sensing," *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.
[2] J. M. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical, statistical and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Feb. 2012.
[3] J. Plaza, E. M. T. Hendrix, I. Garcia, G. Martin, and A. Plaza, "On endmember identification in hyperspectral images without pure pixels: A comparison of algorithms," *J. Math. Imag. Vis.*, vol. 42, no. 2/3, pp. 163–175, 2012.
[4] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 101–118, 2014.
[5] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 9, pp. 2025–2041, Sep. 2002.
[6] D. M. Rogge, B. Rivard, J. Zhang, A. Sanchez, J. Harris, and J. Feng, "Integration of spatial-spectral information for the improved extraction of endmembers," *Remote Sens. Environ.*, vol. 110, no. 3, pp. 287–303, 2007.
[7] M. Zortea and A. Plaza, "Spatial preprocessing for endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2679–2693, Aug. 2009.
[8] G. Martin and A. Plaza, "Region-based spatial preprocessing for endmember extraction and spectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 4, pp. 745–749, Apr. 2011.
[9] G. Martin and A. Plaza, "Spatial-spectral preprocessing prior to endmember identification and unmixing of remotely sensed hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 380–395, Feb. 2012.
[10] J. M. P. Nascimento, J. M. Bioucas-Dias, J. M. Rodriguez Alves, V. Silva, and A. Plaza, "Parallel hyperspectral unmixing on GPUs," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 666–670, Mar. 2013.
[11] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.
[12] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*. Berlin, Germany: Springer-Verlag, 1999.
[13] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Proc. JPL Airborne Earth Sci. Workshop*, 1995, pp. 23–26.
[14] S. Sánchez, R. Ramalho, L. Sousa, and A. Plaza, "Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs," *J. Real-Time Image Process.*, vol. 10, no. 3, pp. 469–483, 2015.