HyperMix: An Open-Source Tool for Fast Spectral Unmixing on Graphics Processing Units

Luis Ignacio Jiménez and Antonio Plaza, *Fellow, IEEE*

***Abstract*—Spectral unmixing has been a popular technique for analyzing remotely sensed hyperspectral images. The goal of unmixing is to find a collection of pure spectral constituents (called *endmembers*) that can explain each (possibly mixed) pixel of the scene as a combination of endmembers, weighted by their coverage fractions in the pixel or *abundances*. Over the last years, many algorithms have been presented to address the three main parts of the spectral unmixing chain: 1) estimation of the number of endmembers; 2) identification of the endmember signatures; and 3) estimation of the per-pixel fractional abundances. However, to date, there is no standardized tool that integrates these algorithms in a unified framework. In this letter, we present HyperMix, an open-source tool for spectral unmixing that integrates different approaches for spectral unmixing and allows building unmixing chains in graphical fashion, so that the end-user can define one or several spectral unmixing chains in fully configurable mode. HyperMix provides efficient implementations of most of the algorithms used for spectral unmixing, so that the tool automatically recognizes if the computer has a graphics processing unit (GPU) available and optimizes the execution of these algorithms in the GPU. This allows for the execution of spectral unmixing chains on large hyperspectral scenes in computationally efficient fashion. The tool is available online from http://hypercomphypermix.blogspot.com.es and has been validated with real hyperspectral scenes, providing state-of-the-art unmixing results.**

***Index Terms*—Graphics processing units (GPUs), HyperMix, hyperspectral imaging, open-source, spectral unmixing.**

I. Introduction

HYPERSPECTRAL remote sensing is based on the capability of imaging spectrometers to collect reflectance data, along different wavelength bands, for the same area on the surface of the Earth. For instance, the NASA Jet Propulsion Laboratory's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) covers the wavelength range from 0.4 to 2.5 μm (visible and near-infrared spectrum) using 224 spectral channels [1]. A hyperspectral data set can be therefore seen as a cube in which each pixel is given by the spectral signature of the underlying materials in that area of the image.

One of the main issues in hyperspectral analysis is the mixed pixel problem [2], which depends on the spatial resolution of the data and also on the characteristics of the area which is being imaged. To address this problem, spectral unmixing finds a collection of pure spectral constituents (called *endmembers*) that can explain each (possibly mixed) pixel of the scene as a combination of endmembers, weighted by their coverage fractions in the pixel or *abundances* [3].

Over the last years, many algorithms have been presented to address the three main parts of the spectral unmixing chain: 1) estimation of the number of endmembers; 2) identification of the endmember signatures; and 3) estimation of the per-pixel fractional abundances [2]. Two major techniques have been used for spectral unmixing purposes: the linear mixture model, which assumes that the materials are combined linearly, and the nonlinear mixture model, which assumes that there are nonlinear interactions between the endmember substances [4]. The linear model is generally considered more computationally tractable, but in both cases, the complexity and high dimensionality of the hyperspectral scenes bring computational challenges that make spectral unmixing techniques appealing for implementation in high-performance computing systems [5], [6]. For instance, graphics processing units (GPUs) have been widely used to accelerate hyperspectral imaging algorithms [7], [8]. GPUs are a low-weight and low-cost hardware platform in which it is possible to accelerate operations and methods in order to easily obtain better computational performance. The number of processor cores depends of the architecture and the model of the GPU. The possibilities of these units go beyond their price and offer an unprecedented potential to accelerate hyperspectral imaging problems.

Despite the popularity of hyperspectral unmixing techniques and their high computational demands, to date, there is no standardized tool that allows for the computationally efficient execution of spectral unmixing chains in a unified, graphical, and fully configurable framework. In this letter, we describe HyperMix, an open-source tool for spectral unmixing which integrates different approaches for spectral unmixing and allows building unmixing chains in graphical and fully configurable fashion, allowing an end-user to intuitively define the characteristics of spectral unmixing chains for hyperspectral analysis applications. In previous developments [9], the tool included several unmixing algorithms covering the different parts of the unmixing chain. A main innovation presented in this letter is the capability of the tool to define unmixing chains in graphical fashion and to automatically recognize if the computer has a GPU available, in which case the execution of the algorithm is optimized for the available GPU device.

Manuscript received March 26, 2015; revised May 6, 2015; accepted May 15, 2015. This work was supported by "Tools for Open Multi-Risk Assessment using Earth Observation Data" (TOLOMEO) under Marie Curie International Research Staff Exchange Scheme (PIRSES-GA-2009).

The authors are with the Hyperspectral Computing Laboratory, University of Extremadura, 10071 Cáceres, Spain (e-mail: luijimenez@unex.es; aplaza@unex.es).

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.2015.2435001

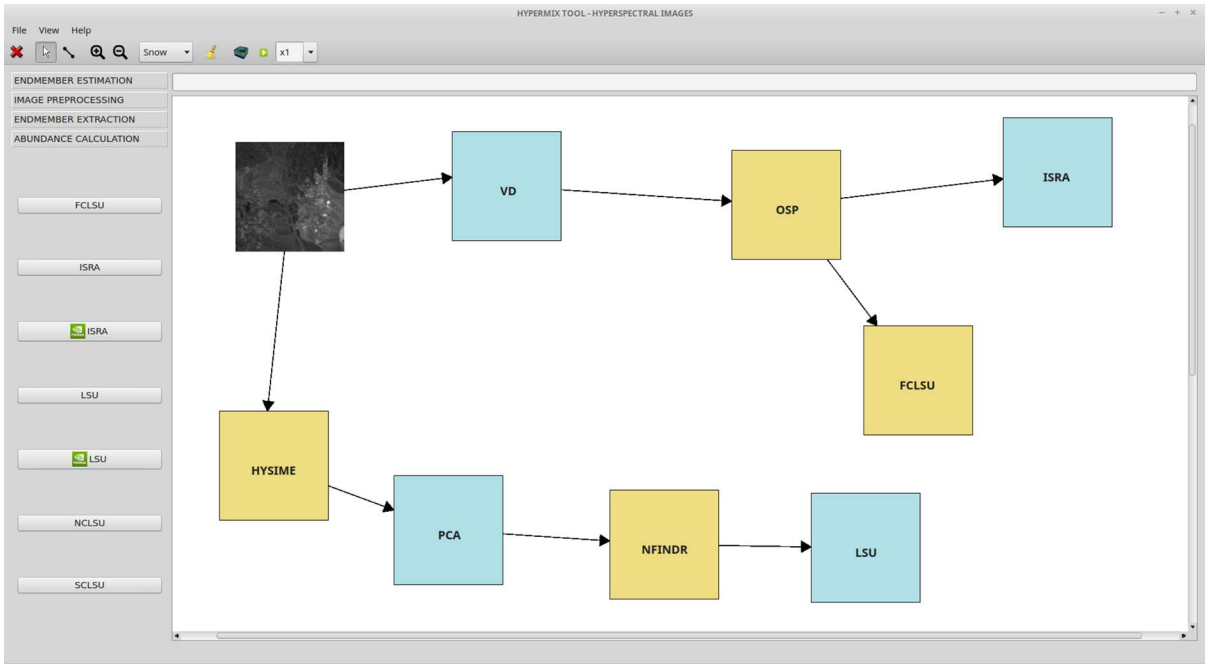

Fig. 1. Flowchart diagram canvas in HyperMix, illustrating the construction of several full spectral unmixing chains for a hyperspectral image collected by AVIRIS over the Cuprite mining district, Nevada.

II. HyperMix Tool

The HyperMix tool was developed following the open-source philosophy for Linux and Windows platforms. The tool is designed to be easy to use; hence, it is also suitable for educational purposes. Originally, the tool was mainly intended to visualize algorithm outcomes and to offer an easy way to integrate spectral unmixing methods. However, the interface of the tool has been completely rebuilt, and the tool now includes the capability to compose spectral unmixing chains and integrate new spectral unmixing algorithms in a straightforward manner. This is accomplished by means of a flowchart diagram canvas (see Fig. 1) developed in order to allow the end-user to configure a desired unmixing chain in a flexible manner. In other words, the user can create a complete hyperspectral unmixing chain (or several ones at the same time), and the outcomes of each individual stage can be used as inputs to the subsequent stages. Each method (or hyperspectral image) is considered as an object or component that the user can manage independently. Related to this dynamic construction of chains is the visualization of the results provided by them. The tool also allows us to load/store/visualize hyperspectral images and endmember/abundance results using different views, as illustrated in Fig. 2.

HyperMix has been designed as an open-source application for hyperspectral data processing and evaluation. For this purpose, among other possibilities considered, we developed the tool using the cross-platform application framework `Qt`,[1] which uses the standard `C++` programming language with extensions that simplify the handling of events required to implement an easy and powerful tool. Moreover, the `Qt` library can be used by other programming languages via binding languages. This makes it possible to expand HyperMix functionalities by including advanced database management or network support. Presently, HyperMix is able to manage the great memory requirements imposed by high-dimensional hyperspectral images using external binary files in band sequential (BSQ) format, which can be read by the main program, and also managing independent external binary files which implement the different unmixing methods (treated as components). It should be noted that the amount of memory needed to store the images through different data types is limited by the operating system (OS) where the tool is running. While Linux generally has no limits, Windows OS has different limitations depending which version is used. In this regard, our tool has been designed to handle the processing algorithms as external operators that communicate with the main tool through binary files, so that the memory management for each of them depends directly on how the tool is implemented. The main program calls these external operators in a proper order, as indicated by the workflow defined in the diagram canvas. The external binaries can be implemented in different programming languages. This has the advantage that we can easily incorporate new processing algorithms that can be executed as independent processes. In the current version of the tool, we use the `C++` programming language for the serial implementations and the NVidia Compute Unified Device Architecture (CUDA)[2] for the GPU versions, leaving open the possibility to include methods developed in other languages, such as OpenCL,[3] or different specific libraries such as the basic linear algebra subprograms (BLAS).[4] This design strategy provides great flexibility about machine requirements, as well as opportunities to create a community of users able to share and exchange different methods.

The spectral unmixing methods currently included in the tool are the following ones. For the identification of the number of endmembers, the virtual dimensionality (VD) [10] and

[1] http://qt-project.org/

[2] https://developer.nvidia.com/cuda-zone

[3] https://www.khronos.org/opencl

[4] http://www.netlib.org/blas

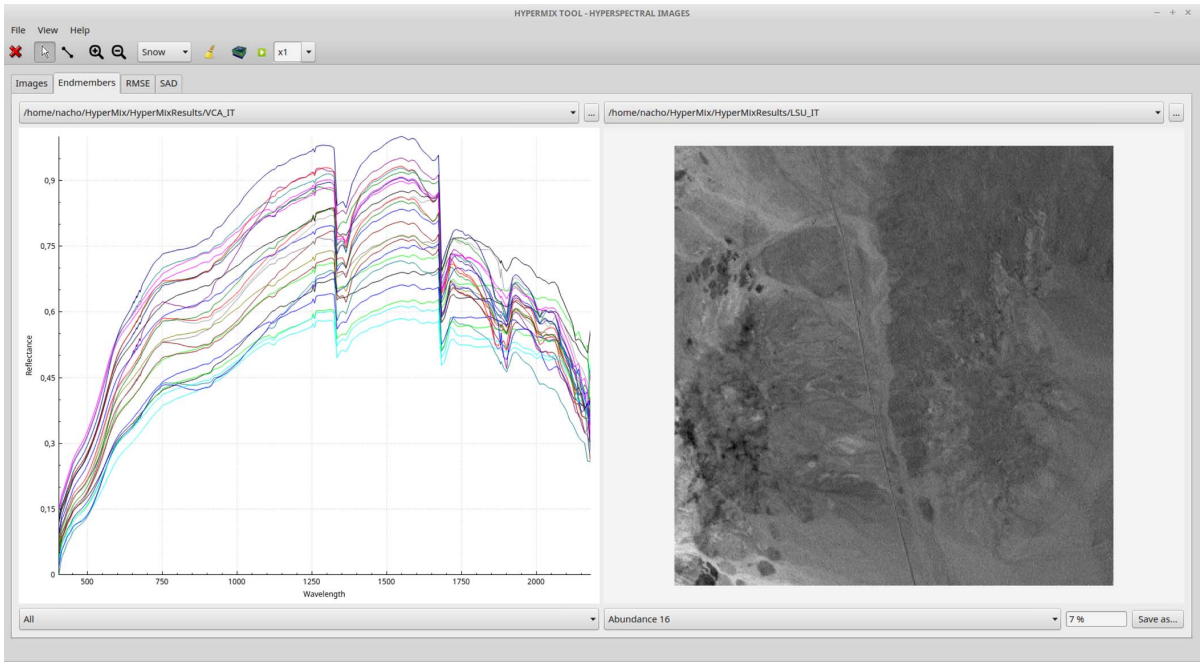

Fig. 2. Visualization of endmembers and abundances, extracted using HyperMix from the AVIRIS Cuprite hyperspectral image.

TABLE I
HYPERSPECTRAL UNMIXING METHODS CURRENTLY INCLUDED IN HYPERMIX (THE GPU IMPLEMENTATIONS OF THESE METHODS HAVE BEEN DESCRIBED IN [20]–[22])

Method	Serial version	GPU version	Reference
VD	✓	✓	[10]
HYSIME	✓		[11]
PCA	✓	✓	[12]
SPP	✓	✓	[13]
OSP	✓		[14]
IEA	✓	✓	[16]
NFINDR	✓	✓	[15]
VCA	✓	✓	[17]
LSU	✓	✓	[18]
SCLSU	✓		[18]
NCLSU	✓		[18]
FCLSU	✓		[18]
ISRA	✓	✓	[19]

the hyperspectral subspace identification with minimum error (HySime) [11] are available. The tool also includes methods for dimensionality reduction and preprocessing, such as the principal component analysis (PCA) [12] and spatial preprocessing (SPP) [13]. For endmember extraction purposes, the tool includes methods such as the orthogonal subspace projection (OSP) [14], NFINDR [15], iterative error analysis (IEA) [16], and vertex component analysis (VCA) [17]. Finally, for the abundance estimation step, the tool includes the least squares unmixing (LSU) method, the sum-to-one constrained LSU (SCLSU), the nonnegativity constrained LSU (NCLSU), the fully constrained LSU (FCLSU) [18], and the image space reconstruction algorithm (ISRA) [19]. Table I summarizes the currently available algorithms in HyperMix, including the serial and GPU versions.

A main feature of the tool is the possibility to run the aforementioned algorithms in parallel in an NVidia GPU if the machine in which the tool is installed has one available, thus allowing us to significantly reduce the computational complexity of the algorithms. The tool automatically recognizes if the system has a GPU available or not. The configuration parameters for each implemented method are made in the codification itself, i.e., we automatically allocate the number of resources (e.g., compute threads, grids, etc.) based on the characteristics of the hardware platform recognized. The techniques currently included in the tool have been tested under CUDA 6.X compatibility. The recognition of the machine capability to run GPU operators is automatically obtained during the installation of the tool, so if there is no such possibility (i.e., the system has no GPU), only the serial versions will be installed. The GPU implementations of these algorithms have been extensively described in [20]–[22]. Another important advantage of HyperMix is that it is very easy to include new methods to the current set of operators. In other words, any user can produce his/her own algorithms (using different programming languages) and include them in the tool as binary files, thus simplifying the procedure to create new content and updates and allowing other users to share new developments. A blog has also been created in the tool Website to foster interactions between HyperMix users.

Other important features included in HyperMix are those oriented to evaluate the accuracy of the results provided by the considered spectral unmixing chains. For this purpose, the tool incorporates the root-mean-square error (rmse) and the normalized rmse (NRMSE) [23], both frequently used in order to evaluate the quality of the extracted endmembers/abundances by comparing the original hyperspectral image and a reconstructed version using the estimated endmembers and abundances. The tool also allows using the spectral angle distance (SAD) [23] to compare the spectral signatures of the extracted endmembers with regard to reference signatures, possibly available *a priori* in a spectral library.

III. EXPERIMENTAL RESULTS

Our experiments to evaluate the tool are intended to analyze every step of a hyperspectral unmixing chain and also to test the

TABLE II
MEAN EXECUTION TIMES (IN SECONDS) FOR THE SERIAL AND GPU IMPLEMENTATIONS OF THE DIFFERENT PARTS OF THE CONSIDERED SPECTRAL UNMIXING CHAIN AFTER TEN MONTE CARLO RUNS. THE SPEEDUPS FOR EACH PART OF THE CHAIN ARE REPORTED IN THE PARENTHESES

Image	Device	VD	SPP	VCA	LSU	Full Chain
Cuprite	CPU	5.162	5.950	12.753	0.752	24.617
	GTX580	0.263 (19.627x)	0.291 (20.446x)	0.948 (13.452x)	0.160 (4.7x)	1.662 (14.820x)
	GT740M	0.784 (6.584x)	1.170 (5.085x)	1.721 (7.410x)	0.244 (3.081x)	3.919 (6.281x)
World Trade Center	CPU	18.923	41.474	49.991	3.126	113.514
	GTX580	0.574 (32.96x)	0.824 (50.33x)	4.254 (11.75x)	0.279 (11.204x)	5.931 (19.142x)
	GT740M	1.912 (9.89x)	3.411 (12.15x)	5.605 (9.919x)	0.507 (6.165x)	11.435 (9.926x)

TABLE III
SPECTRAL SIMILARITY SCORES OBTAINED USING TWO MATCHING ALGORITHMS AVAILABLE IN HYPERMIX BASED ON THE SAD DISTANCE (IN DEGREES). THE COMPARISON IS CONDUCTED BETWEEN THE ENDMEMBERS EXTRACTED BY A FULL SPECTRAL UNMIXING CHAIN WITH REGARD TO SEVERAL USGS LIBRARY SIGNATURES FOR FIVE HIGHLY REPRESENTATIVE MINERALS IN THE CUPRITE MINING DISTRICT

Matching algorithm	Alunite	Buddingtonite	Calcite	Kaolinite	Muscovite	Mean
Minimum SAD	11.894	7.271	10.743	13.946	5.237	9.820
Average SAD	11.894	7.643	10.594	13.946	5.237	9.861

different options that the tool offers. For that purpose, in this letter, we use two hyperspectral scenes (a database of synthetic scenes [24] is also included with the tool distribution). The first one is the well-known AVIRIS Cuprite data set, available online in reflectance units[5] after atmospheric correction. This scene has been widely used to evaluate the performance of hyperspectral unmixing algorithms. The scene contains 224 spectral bands in the range of 0.4 to 2.5 μm with 350 $\times$ 350 pixels and spatial resolution of 20 m per pixel. Several bands were removed due to water absorption and low SNR in those bands, leaving 188 reflectance channels. Fig. 2 shows an example of how the image is processed using HyperMix. The Cuprite site has several exposed minerals included in the USGS spectral library,[6] and we use several representative mineral signatures to evaluate the quality of the extracted endmembers. The other scene used in experiments was collected by AVIRIS over the World Trade Center (WTC) area in New York City. This image was obtained after the terrorist attacks in 2001. The data set consists of 614 $\times$ 512 pixels and 224 spectral bands, with a spatial resolution of 1.7 m per pixel. The image is available in radiance units and is generally regarded as a benchmark for evaluating parallel implementations of hyperspectral image processing algorithms.

To illustrate the HyperMix tool, we constructed a full spectral unmixing chain (using the canvas feature) and evaluated its performance using the serial and GPU implementations available in the tool. For the considered chain, we used VD for identifying the number of endmembers, SPP for spatial preprocessing, VCA for endmember extraction, and LSU for abundance estimation. The parameters were set empirically as follows: for the VD, we used a false alarm probability $P_F = 10^{-3}$ [10], and for the SPP, we used a spatial window size of $ws = 7$ [13]. The remaining blocks of the spectral unmixing chain did not require any input parameters. The experiments were conducted using two different GPUs: one for desktop machines (NVidia GeForce GTX580 with 512 cores and 1536 MB GDDR5) and another one for laptop devices (NVidia GeForce GT740M with 384 cores and up to 2 GB GDDR3). The computational times measured in the GPUs were compared against those obtained in an Intel Core i7 CPU 920 @ 2.67 GHz $\times$ 8 with 6 GB of memory. The accuracy of the obtained results was measured using SAD for the AVIRIS Cuprite scene (using the USGS spectral library of reference mineral signatures) and the NRMSE for the two considered scenes, using this metric as an indicator of how accurately the reconstructed version each image (obtained after applying the linear mixture model) approximates the original one.

At this point, it is important to note that all of the results reported were obtained after conducting ten Monte Carlo runs of the same chain for each image, using independent methods implemented in serial and parallel. As shown in Table II, the performance obtained for the whole chain for the NVidia GTX580 is quite remarkable, obtaining a total speedup of 14.820$\times$ for the Cuprite scene and 19.142$\times$ for the WTC scene. For instance, this allowed us to unmix the WTC scene in just 5.931 s in the GTX580 (as opposed to 113.514 s in the CPU). Although the speedups reported for the Cuprite scene are lower, this is related to the smaller size of the image (which can be unmixed in just 1.662 s in the GTX580). The speedups achieved by each part of the chain vary, but for instance, remarkable speedups were obtained for the VD (32.96$\times$ when processing the WTC scene in the GTX580) and the SPP (50.33$\times$ speedup in the GTX580). The results obtained in the GT740M are also quite remarkable given the fact that this GPU is available for laptop computers. Overall, we can conclude from Table II that the GPU feature in HyperMix allows significantly improved computational performance.

To evaluate the accuracy of the extracted endmembers from the Cuprite image, we have used the HyperMix tool to obtain the SAD scores between the endmembers extracted by the previously considered unmixing chain and some reference mineral signatures contained in the USGS spectral library (specifically, we have considered the minerals: alunite, buddingtonite, calcite, kaolinite, and muscovite which appear prominently in the Cuprite mining district). Two matching algorithms are used to compare the set of extracted endmembers with regard to the reference USGS signatures. The first one (minimum SAD) establishes the minimum SAD between the reference signatures and the endmembers by comparing them one by one

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

without checking if a different ordering of the endmembers or the reference signatures when performing the matching would improve the results. The second method (average SAD) performs the matching after a number of executions of the minimum SAD algorithm by randomly sorting the endmembers and the reference signatures, thus making sure that a more optimal matching is achieved. Table III shows the SAD scores (in degrees) obtained by the HyperMix tool after comparing the extracted endmembers with the reference USGS signatures. As shown by Table III, the extracted endmembers are very similar, spectrally, with regard to the reference USGS signatures. An evaluation of the NRMSE scores obtained using the HyperMix tool after comparing the reconstructed versions of the two considered hyperspectral scenes also reveals high similarity scores, with values of 0.0185 for the Cuprite scene and 0.0058 for the WTC scene using the considered unmixing chain.

IV. Conclusion And Future Lines

In this letter, we have presented an open-source tool called HyperMix which contains a variety of algorithms (iterative and parallel) for spectral unmixing of remotely sensed hyperspectral data sets. The tool offers an easy way to manage these algorithms and build spectral unmixing chains, along with comprehensive options to display and validate the obtained unmixing results. The developed open-source tool and quantitative comparison of unmixing algorithms is expected to be of great interest to both algorithm developers and end-users of spectral unmixing techniques, as this is the first tool of its kind in the area of spectral unmixing. In addition to the inclusion of additional techniques for spectral unmixing (which is very easy to manage in the current version), our future lines of research will be focused toward offering remote access to a repository of hyperspectral scenes [24] through HyperMix and extending the capability of HyperMix to include other kinds of techniques for hyperspectral image processing. We will maintain and expand the tool with additional versions compatible with new GPU families.

References

[1] R. O. Green *et al.*, "Imaging spectroscopy and the Airborne Visible/ Infrared Imaging Spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.

[2] J. 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, Apr. 2012.

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

[4] R. Heylen, M. Parente, and P. Gader, "A review of nonlinear hyperspectral unmixing methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1844–1868, Jun. 2014.

[5] A. Plaza, Q. Du, Y.-L. Chang, and R. King, "High performance computing for hyperspectral remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 528–544, Sep. 2011.

[6] C. Lee, S. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 508–527, Sep. 2011.

[7] A. Plaza, J. Plaza, A. Paz, and S. Sánchez, "Parallel hyperspectral image and signal processing," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 119–126, May 2011.

[8] S. Sánchez, A. Paz, G. Martin, and A. Plaza, "Parallel unmixing of remotely sensed hyperspectral images on commodity graphics processing units," *Concurr. Comput. Pract. Exp.*, vol. 23, no. 13, pp. 1538–1557, Sep. 2011.

[9] L.-I. Jimenez, G. Martin, and A. Plaza, "HyperMix: A new tool for quantitative evaluation of endmember identification and spectral unmixing techniques," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 1393–1396.

[10] C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 608–619, Mar. 2004.

[11] J. M. Bioucas-Dias and J. M. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.

[12] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, vol. 3. Berlin, Germany: Springer-Verlag, 1999.

[13] M. Zortea and A. Plaza, "Spatial preprocessing for endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2679–2693, Aug. 2009.

[14] J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.

[15] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Proc. SPIE Int. Soc. Opt. Photon.*, 1999, pp. 266–275.

[16] R. Neville, K. Staenz, T. Szeredi, J. Lefebvre, and P. Hauff, "Automatic endmember extraction from hyperspectral data for mineral exploration," in *Proc. 21st Can. Symp. Remote Sens*, 1999, pp. 21–24.

[17] J. M. Nascimento and J. M. Bioucas Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.

[18] D. C. Heinz and C.-I. Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.

[19] M. E. Daube-Witherspoon and G. Muehllehner, "An iterative image space reconstruction algorithm suitable for volume ECT," *IEEE Trans. Med. Imag.*, vol. MI-5, no. 2, pp. 61–66, Jun. 1986.

[20] S. Sanchez, R. Ramalho, L. Sousa, and A. Plaza, "Real-time implementation of hyperspectral image unmixing on GPUs," *J. Real-Time Image Process.*, to be published.

[21] S. Sanchez and A. Plaza, "Fast determination of the number of endmembers for real-time hyperspectral unmixing on GPUs," *J. Real-Time Image Process.*, vol. 9, no. 3, pp. 397–405, Sep. 2014.

[22] J. Nascimento, J. Bioucas-Dias, J. 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. 2014.

[23] N. Keshava and J. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.

[24] J. Sevilla and A. Plaza, "A new digital repository for hyperspectral imagery with unmixing-based retrieval functionality implemented on GPUs," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2267–2280, Jun. 2014.