the United States and a university and aims in building such tools. Works of art are dynamic entities and express a deep communication between the artist and viewer that does not need to be translated into words. It is direct and inspiring. The more we understand the visual message, the more we come to appreciate the artist's particular depth of expression. That is why art historians and conservators, with the aid of scientists, are always looking for new ways of getting under the skin and clarifying a work of art.

With this article we hope to invigorate research in the digital conservation of artworks and inspire a new generation of engineering students in tackling such problems that can bridge the world of science with the arts.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the Andrew W. Mellon Foundation. The authors would like to thank Prof. Robert Erdmann of the University of Arizona in Tucson for making available his registered images of Druet and modern state. They would also like to thank Frank Zuccari, Grainger executive director of conservation at AIC, conservators (in particular Kelly Keegan and Lucia Bay), art historians, and imaging specialists at AIC that participated in this effort. Finally they would like to acknowledge the assistance of Jana Zujovic, Kai Hayashi, Benjamin Prosnitz, and Anna Targowska (students at Northwestern University at the time).

AUTHORS

Sotirios A. Tsaftaris (s-tsaftaris@northwestern.edu) is a research assistant professor with the Department of Electrical Engineering and Computer Science and the Department of Radiology, Feinberg School of Medicine at Northwestern University.

Kristin H. Lister (klister@artic.edu) is a paintings conservator at the Art Institute of Chicago where she specializes in modern and contemporary paintings.

Inge Fiedler (ifiedler@artic.edu) is the conservation microscopist at the Art Institute of Chicago.

Francesca Casadio (fcasadio@artic.edu) is the Andrew W. Mellon senior conservation scientist at the Art Institute of Chicago.

Aggelos K. Katsaggelos (aggk@eecs.northwestern.edu) is currently a professor with the Department of Electrical Engineering and Computer Science at Northwestern University.

REFERENCES

[1] S. D'Alessandro and J. Elderfield, *Matisse: Radical Invention, 1913-1917.* Chicago, IL: Art Institute of Chicago, 2010.

[2] Art Institute of Chicago. (25 Aug 2010). Matisse: Radical Invention 1913–1917. Exhibition Web site [Online]. Available: http://www.artic.edu/aic/exhibitions/matisse/splash.html

[3] M. Barni, A. Pelagotti, and A. Piva, "Image processing for the analysis and conservation of paintings: Opportunities and challenges," *IEEE Signal Processing Mag.*, vol. 22, no. 5, pp. 141–144, 2005.

[4] R. S. Berns, S. Byrns, F. Casadio, I. Fiedler, C. Gallagher, F. H. Imai, A. Newman, and L. A. Taplin, "Rejuvenating the color palette of Georges Seurat's a Sunday on La Grande Jatte—1884: A simulation," *Color Res. Appl.*, vol. 31, no. 4, pp. 278–293, 2006.

[5] J. Glausiusz, "Matisse's methods revealed," *Nature*, vol. 464, no. 7288, pp. 493–494, Mar. 2010.

[6] G. Burns. (2008). Colorization. *Encyclopedia of Television.* [Online]. Museum of Broadcast Communications, Chicago, IL. Available: http://www.museum.tv/archives/etv/index.html

[7] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 277–280, 2002.

[8] D. Sýkora, J. Buriánek, and J. Žára, "Unsupervised colorization of black-and-white cartoons," in *NPAR '04: Proc. 3rd Int. Symp. Non-Photorealistic Animation and Rendering*, 2004, pp. 121–127.

[9] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[10] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Processing,* vol. 15, no. 5, pp. 1120–1129, 2006.

[11] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y. Xu, and H. Shum, "Natural image colorization," in *Proc. Eurographics Symp. Rendering Techniques,* Grenoble, France, 2007.

[12] M. Styner, C. Brechbuhler, G. Szckely, and G. Gerig, "Parametric estimate of intensity inhomogeneities applied to MRI," *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 153–165, 2000.

[13] I. J. Cox, S. Roy, and S. L. Hingorani, "Dynamic histogram warping of image pairs for constant image brightness," in *Proc. IEEE Int. Conf. Image Processing*, 1995, vol. 2, pp. 366–369.

[SP]

Antonio Plaza, Javier Plaza,
Abel Paz, and Sergio Sánchez

Parallel Hyperspectral Image and Signal Processing

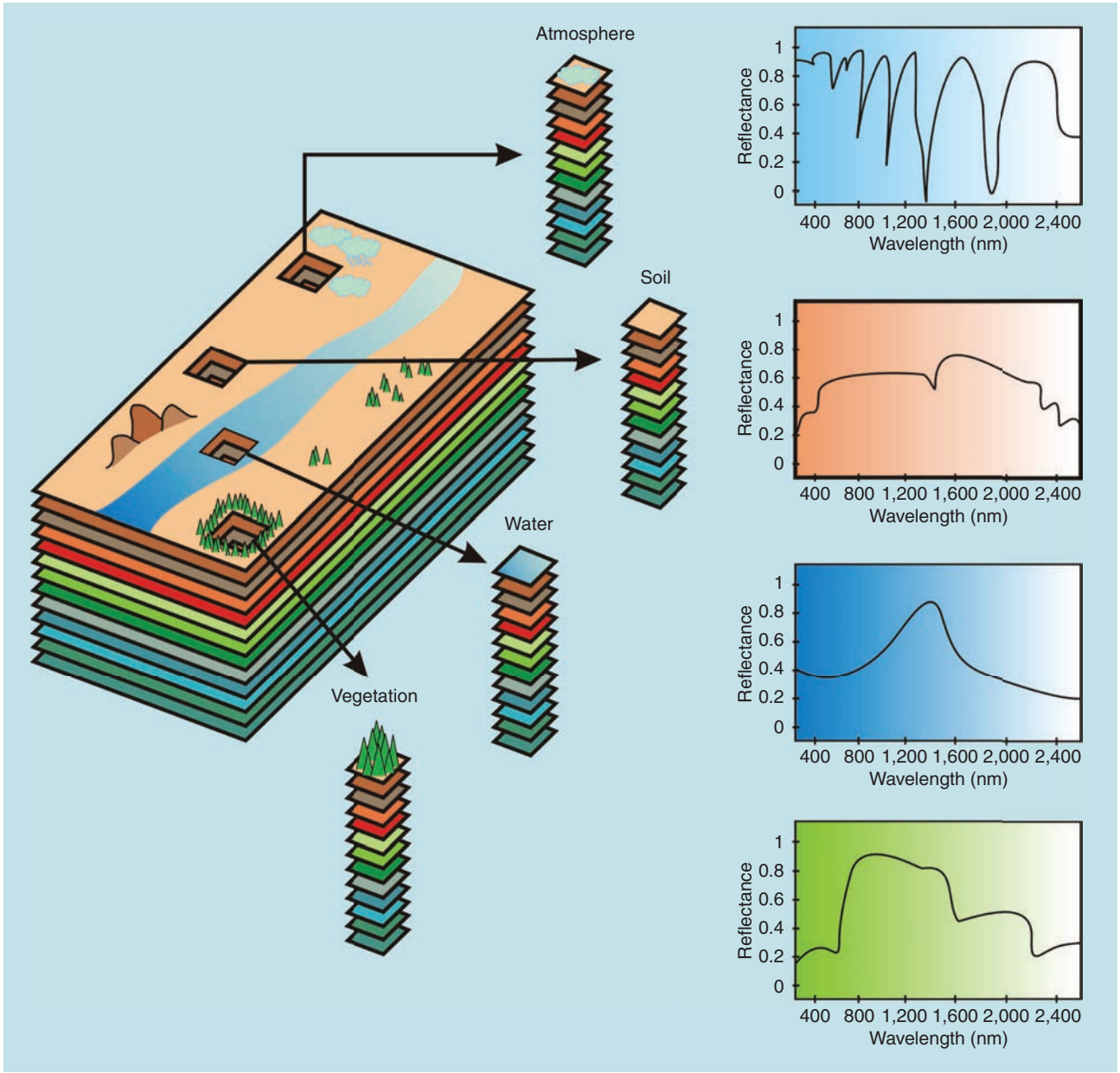
Remotely sensed hyperspectral imaging instruments are capable of collecting hundreds of images corresponding to different wavelength channels for the same area on the surface of the Earth. For instance, NASA is continuously gathering high-dimensional image data with instruments such as the Jet Propulsion Laboratory's Airborne Visible-Infrared Imaging Spectrometer (AVIRIS). This advanced sensor for Earth observation records the visible and near-infrared spectrum of the reflected light using more than 200 spectral bands, thus producing a stack of images in which each pixel (vector) is represented by a spectral signal that uniquely characterizes the underlying objects. The resulting data volume typically comprises several

Digital Object Identifier 10.1109/MSP.2011.940409
Date of publication: 19 April 2011

gigabytes per flight. In this article, we describe the state of the art in the development and application of image and signal processing techniques for advanced information extraction from hyperspectral data. The article also describes new trends for efficient processing of such data using parallel and distributed processing techniques in the context of time-critical applications.

PROBLEM

Hyperspectral imaging is an emerging and fast-growing area in remote sensing. It is concerned with the measurement, analysis, and interpretation of spectra acquired from a given scene (or specific object) at a short, medium, or long distance by an airborne or satellite sensor [1]. The wealth of spectral information available from latest generation hyperspectral imaging instruments (which have substantially increased their spatial, spectral, and temporal resolutions) has quickly introduced new processing challenges. For instance, NASA's AVIRIS (http://aviris.jpl.nasa.gov) is now able to record the visible and near-infrared spectrum (wavelength region from 400 to 2,500 nm) of the reflected light of an area 2–12 km wide and several kilometers long, using 224 spectral bands. The resulting data volume can be seen as a data cube with two spatial and one spectral dimension (see Figure 1). Although AVIRIS is a widely used platform, it constitutes only one source of hyperspectral data. Table 1 summarizes other international Earth observation missions with hyperspectral sensors already launched or to be launched in the near future. While in this work our focus is on



[FIG1] **Hyperspectral data cube.**

[TABLE 1] OVERVIEW OF SOME PRESENT AND FUTURE REMOTE SENSING MISSIONS INCLUDING HYPERSPECTRAL SENSORS.

	HYPERION*	PRISMA†	ENMAP‡	HYSPIRI§
COUNTRY OF ORIGIN	USA	ITALY	GERMANY	USA
SPATIAL RESOLUTION	30 M	5–30 M	30 M	60 M
REVISIT TIME	16 DAYS	3/7 DAYS	4 DAYS	18 DAYS
SPECTRAL RANGE	400–2500 NM	400–2500 NM	420–2450 NM	380–2500 NM
SPECTRAL RESOLUTION	10 NM	10 NM	6.5–10 NM	10 NM
SWATH WIDTH	7.7 KM	30 KM	30 KM	120 KM
EARTH COVERAGE	PARTIAL	FULL	FULL	FULL
LAUNCH	2000	2010	2012	2018
LIFETIME	10 YEARS	≈ 6 YEARS	≈ 6 YEARS	≈ 6 YEARS

*http://eo1.gsfc.nasa.gov †http://www.asi.it/en/flash_en/observing/prisma ‡http://www.enmap.org §http://hyspiri.jpl.nasa.gov

remote sensing applications, hyperspectral sensors have been widely used in many other areas. For instance, hyperspectral cameras are now routinely used for industrial quality control, food inspection, forensics, and medical imaging purposes. Hyperspectral microscopes are also gaining popularity in applications such as nanotoxicology, chemometrics, and drug discovery.

HYPERSPECTRAL IMAGE AND SIGNAL PROCESSING TASKS

The number and variety of processing tasks in hyperspectral remote sensing is enormous [2]. However, the majority of algorithms can be organized according to the following specific tasks [3]:

- *Dimensionality reduction* consists of reducing the dimensionality of the input hyperspectral scene to facilitate subsequent processing tasks.
- *Target and anomaly detection* consists of searching the pixels of a hyperspectral data cube for "rare" (either known or unknown) spectral signatures.
- *Change detection* consists of finding the "significant" (i.e., important to the user) changes between two hyperspectral scenes of the same geographic region.
- *Classification* consists of assigning a label (class) to each pixel of a hyperspectral data cube.
- *Spectral unmixing* consists of estimating the fraction of the pixel area covered by each material present in the scene.

In particular, spectral unmixing has been an alluring exploitation goal since the earliest days of hyperspectral image and signal processing [4]. No matter the spatial resolution, the spectral signatures collected in natural environments are invariably a mixture of signatures of the various materials found within the spatial extent of the ground instantaneous field view of the imaging instrument. For instance, the pixel vector labeled as "vegetation" in Figure 1 may actually comprise a mixture of vegetation and soil or different types of soil and vegetation canopies. In this case, several spectrally pure signatures (called endmembers in hyperspectral imaging terminology) are combined into the same (mixed) pixel.

COMPUTATIONAL REQUIREMENTS

The increased spatial, spectral, and temporal resolution of advanced Earth observation instruments, together with the real-time requirements of certain applications, have led to the incorporation of high-performance computing infrastructure to accelerate spectral signature-related computations [5] (see Figure 2). The wealth of information available from hyperspectral imaging instruments has opened ground-breaking perspectives in several applications, including environmental modeling and assessment for Earth-based and atmospheric studies, risk/hazard prevention and response including wild land fire tracking, biological threat detection, monitoring of oil spills and other types of chemical contamination, target detection for military and defense/security purposes, and urban planning and management studies, among many others [6].

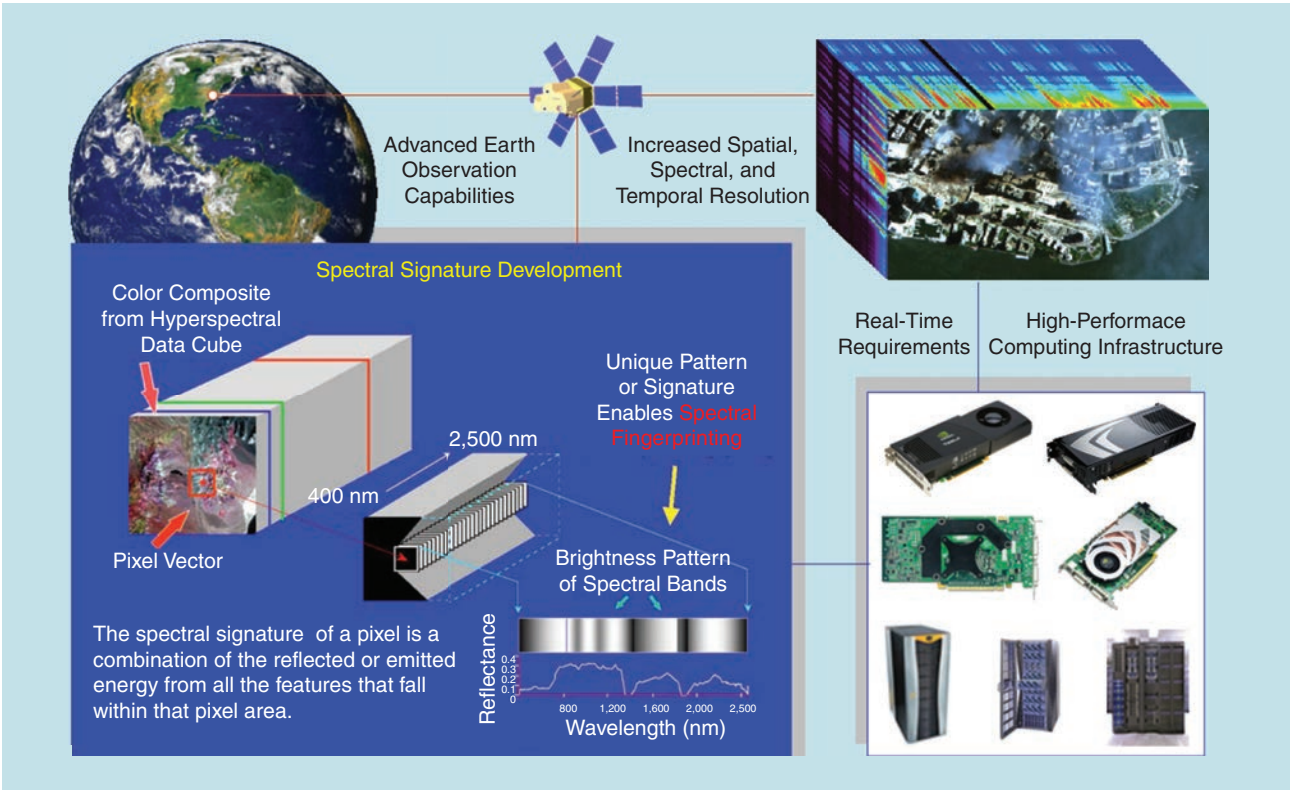
In the following, we review different strategies to unmix hyperspectral data cubes including parallel and distributed processing implementations. An application of the discussed techniques to the analysis of hyperspectral data collected by AVIRIS over the World Trade Center (WTC) in New York City, five days after the terrorist attacks of September 11, 2001, is presented as an application case study.

SPECTRAL UNMIXING OF HYPERSPECTRAL DATA

Spectral unmixing involves the separation of a pixel spectrum into its pure component endmember spectra and the estimation of the abundance value for each endmember in each pixel of the scene [4]. A standard technique for this purpose is linear spectral unmixing, which assumes that the collected spectra at the spectrometer can be expressed in the form of a linear combination of endmembers weighted by their corresponding abundances. Under the linear mixture model assumption, each n-dimensional pixel vector of the hyperspectral scene can be modeled using the following expression [7]:

$$\mathbf{f}_i = \sum_{j=1}^{p} \mathbf{e}_j \cdot \Phi_j + \mathbf{t}, \qquad (1)$$

where $\mathbf{e}_j$ denotes the spectral response of an endmember, Φ_j is a scalar value designating the fractional abundance of the endmember $\mathbf{e}_j$, p is the total number of endmembers, and $\mathbf{t}$ is a noise term. If we assume that $n >> p$, the solution of the problem described in (1) relies on the correct determination of a set $\{\mathbf{e}_j\}_{j=1}^{p}$ of

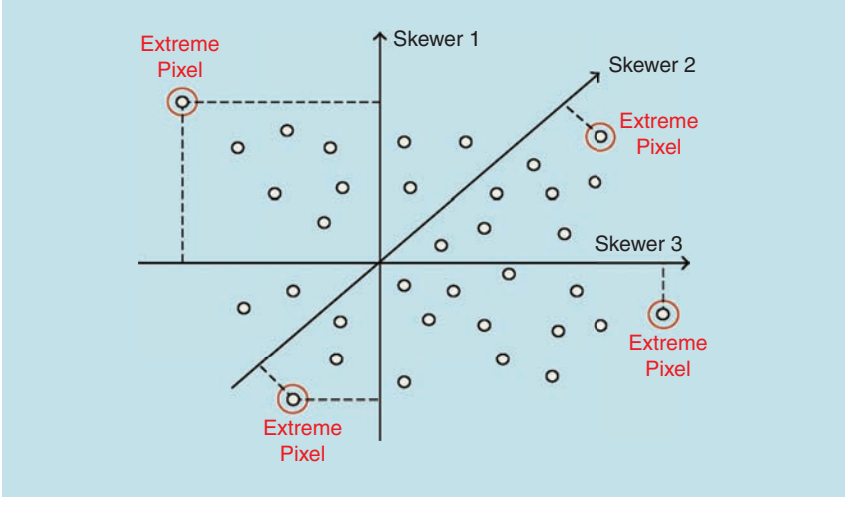


[FIG2] **Computational requirements introduced by hyperspectral imaging instruments.**

endmembers and their correspondent abundance fractions $\Phi = \{\Phi_j\}_{j=1}^{p}$ at each pixel $\mathbf{f}_i$.

ENDMEMBER EXTRACTION

Over the last decade, several algorithms have been developed for the automatic extraction of spectral endmembers directly from the input data [8]. A popular algorithm based on signal processing concepts has been the pixel purity index (PPI) [9], which calculates a spectral purity score for each n-dimensional pixel in the original data by generating random unit vectors (called skewers) so that all pixel vectors are projected onto the skewers and the ones falling at the extremes of each skewer are counted. After many repeated projections to different skewers, those pixels that count above a certain cut-off threshold are declared "pure" (see Figure 3).



[FIG3] **Toy example illustrating the PPI algorithm in a two-dimensional space.**

Another widely used approach to endmember extraction is N-FINDR [10], which begins with a random initial selection of endmembers and iteratively looks for the set of pixels that maximize the volume of the simplex defined by the selected endmembers. This is calculated for every pixel in each endmember position by replacing that endmember and finding the resulting volume. If the replacement results in an increase of volume, the pixel replaces the endmember. This procedure is repeated until there are no more endmember replacements. Other techniques for endmember extraction have been developed based on image processing concepts. Among several others, the automatic morphological endmember extraction (AMEE) uses extended mathematical morphology operators [8] to select spectral endmembers.

ABUNDANCE ESTIMATION

Once a set of endmembers $\mathbf{E} = \{\mathbf{e}_j\}_{j=1}^{p}$ has been obtained, an unconstrained abundance estimation (in least squares

sense) can be obtained by the following expression [7]:

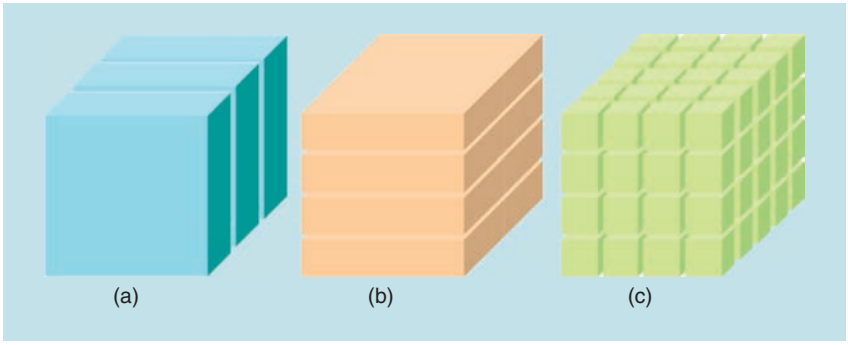
$$\Phi \approx (\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T\mathbf{f}_i. \qquad (2)$$

It should be noted that the abundance estimation in (2) does not satisfy the two physical constraints that are generally imposed in the linear mixture model: the abundance nonnegativity constraint, i.e., $\Phi_j \geq 0$, and the abundance sum-to-one constraint, i.e., $\sum_{j=1}^{p}\Phi_j = 1$. As described in [7], these constraints can be imposed leading to a fully constrained abundance estimation in the framework of linear spectral unmixing.

PARALLEL AND DISTRIBUTED IMPLEMENTATIONS

Several strategies have been explored in the literature to accelerate hyperspectral unmixing algorithms [5]. To take advantage of the computational power offered by parallel computing architectures, a spatial-domain decomposition approach that subdivides the image cube into multiple blocks made up of entire pixel vectors and assigns each block to a different processing element has been adopted [see Figure 4(b)]. The standard hyperspectral unmixing chain (made up of the PPI algorithm for endmember extraction followed by fully constrained abundance estimation) is based on calculations in which pixels are always treated as entire spectral signatures. Therefore, spectral-domain partitioning [see Figure 4(a)] or mixed spatial-spectral partitioning [see Figure 4(c)] are not appropriate for parallel implementation because the calculations made for each hyperspectral pixel would need to originate from several processing elements, thus requiring intensive inter-processor communication.

In our implementation of the unmixing chain for clusters of computers, we adopted a simple master-slave approach in which the master processor distributes spatial-domain partitions of the data to the workers and coordinates their actions. Then, the master gathers the partial results provided by the workers and produces a global result. It is important to emphasize that some steps of this parallel algorithm (e.g., the generation of skewers) are purely sequential. This means that the master node performs some steps of the algorithm on its own. Nevertheless, the execution time of these sequential steps is insignificant in comparison to the total execution time. To adapt this framework to heterogeneous networks of computers, the data partitioning should be made so that each processor executes an amount of work proportional to its speed [11]. For this purpose, we have developed a workload estimation algorithm for heterogeneous networks that assumes that the workload allocated to each processor must be directly proportional to its processing power.



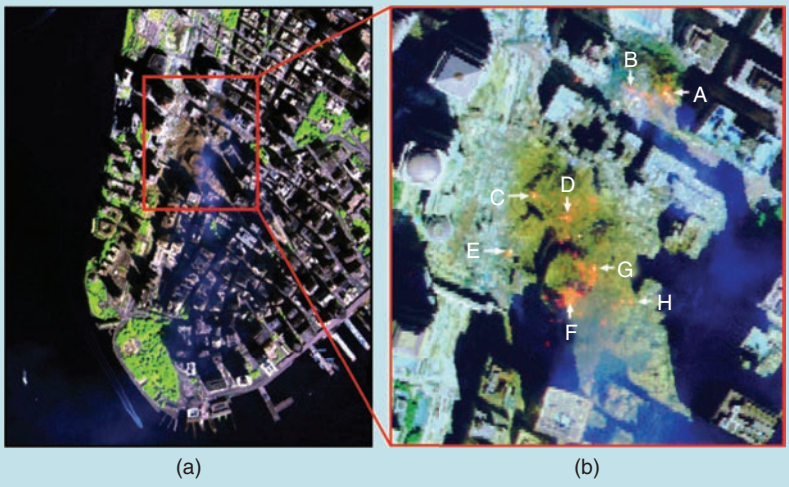
[FIG4] Strategies for hyperspectral data partitioning: (a) spatial, (b) spectral, and (c) spatial-spectral.

Even though hyperspectral processing algorithms map nicely to clusters and heterogeneous networks, these systems are generally expensive and difficult to adapt to on-board data processing scenarios in which low-weight and low-power integrated components are essential to reduce mission payload. To bridge the gap towards on-board, real-time analysis of hyperspectral data, parallel versions of spectral unmixing algorithms have been developed on field programmable gate arrays (FPGAs) and graphic processing units (GPUs) [5]. FPGAs are now fully reconfigurable, thus allowing for adaptive selection of a data processing algorithm (out of a pool of available ones) from a control station on Earth. Our strategy for implementing the hyperspectral unmixing chain in reconfigurable hardware is aimed at enhancing replicability and reusability of slices in FPGA devices through the utilization of a systolic array, which performs the skewer projections in an embarrassingly parallel fashion. A read queue and a transmitter are also used to send the endmembers to the FPGA via an RS232 port, while a control unit and modules for direct memory access and random generation of skewers are also implemented.

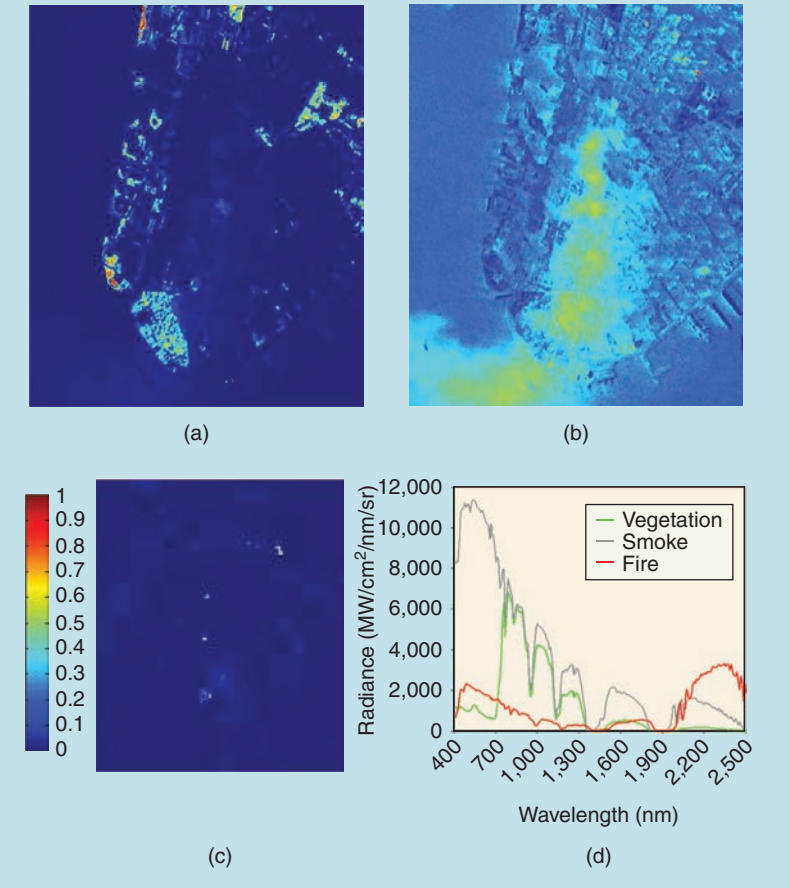
On the other hand, the emergence of GPUs (driven by the ever-growing demands of the video game industry) has allowed these systems to evolve from expensive application-specific units into highly parallel and programmable commodity components. GPUs can be abstracted in terms of a stream model, under which all data sets are represented as streams (i.e., ordered data sets). Our implementation of the hyperspectral unmixing chain is constructed by chaining so-called kernels, which operate on the input data stream, and produce the endmembers and fractional abundances as outputs. Thereby, data-level parallelism is exposed to hardware, and kernels can be concurrently applied with minimal synchronization, thus maximizing parallel performance.

APPLICATION CASE STUDY

To illustrate the potential of hyperspectral technology in an application with real-time constraints, we use an AVIRIS scene collected over the WTC in New York City on 16 September 2001, five days after the terrorist attacks that collapsed the two main towers and other buildings in the WTC complex. The data consists of 614×512 pixels and 224



[FIG5] (a) AVIRIS hyperspectral image collected over the WTC. (b) Location of thermal hot spots in the WTC complex (available online: http://pubs.usgs.gov/of/2001/ofr-01-0429/hotspot.key.tgif.gif).



[FIG6] Abundance maps estimated for (a) vegetation, (b) smoke, and (c) fire. The vegetation and smoke abundance maps correspond to the full scene, while the fire abundance map corresponds to the subscene in the rightmost part of Figure 5. The spectral endmembers extracted from the original scene are also displayed in (d).

bands, for a total size of 140 MB. Figure 5(a) shows a false color composite of the data set using the 1,682, 1,107, and 655 nm channels, displayed as red, green, and blue, respectively. Vegetated areas appear green, while burned areas appear dark gray. Smoke coming from the WTC area (in the red rectangle) appears bright blue due to high spectral reflectance in the 655-nm channel. Figure 5(b) shows a thermal map with the locations of hot spots at the WTC area, available from the U.S. Geological Survey (USGS) (http://speclab.cr.usgs.gov/wtc). This map was used to validate a hyperspectral unmixing chain available in ENVI, a popular commercial software package distributed by ITT Visual Information Solutions (http://www.ittvis.com/ProductServices/ENVI.aspx). The chain consists of two stages: 1) endmember extraction using the PPI algorithm (with 10^4 skewers) and 2) fully constrained abundance estimation. It was implemented on four different parallel platforms:

- Thunderhead, a Beowulf cluster at NASA's Goddard Space Flight Center in Maryland and made up of 256 Intel Xeon nodes at 2.4 GHz, each with 1 GB of memory and 80 GB of main memory, interconnected with 2 Ghz optical fibre Myrinet. This is an expensive platform in terms of cost and maintenance. The parallel implementation was developed using the C++ programming language with calls to MPI library (http://www.mcs.anl.gov/research/projects/mpi).
- A heterogeneous network that consists of 16 different workstations (one i386 Intel Pentium 4, six Intel Xeons, seven AMD Athlons, and a SUNW UltraSparc) interconnected via heterogeneous links [5].
- A Xilinx Virtex-II XC2V6000-6 FPGA, an architecture that is similar to other FPGAs that have been certified by several international agencies for aerospace operation. The cost of this platform is around US$300. The parallel implementation was synthesized using Handel-C (http://www.mentor.com/products/fpga/handel-c), a hardware design and prototyping language.

[TABLE 2] TIMES (SECONDS) MEASURED AFTER PROCESSING THE AVIRIS WTC HYPERSPECTRAL SCENE IN DIFFERENT TYPES OF PARALLEL COMPUTING ARCHITECTURES.

	THUNDERHEAD BEOWULF CLUSTER									HETEROGENEOUS NETWORK
# CPUS	**1**	**4**	**16**	**36**	**64**	**100**	**144**	**196**	**256**	**16**
SEQUENTIAL COMPUTATIONS	1163.05	1.63	1.26	1.12	1.19	1.06	0.84	0.91	0.58	1.69
PARALLEL COMPUTATIONS	0	292.09	73.24	30.46	15.44	8.76	5.08	3.18	1.91	79.56
COMMUNICATIONS	0	2.20	2.41	2.39	2.21	2.46	2.65	2.32	2.49	3.56
TOTAL TIME[‡]	1163.05	295.92	76.91	33.97	18.84	12.38	8.57	6.41	4.98	83.05
SPEEDUP	–	3.93	15.12	34.23	61.73	93.89	135.67	181.34	233.45	13.23

[‡]The total time consumed by our FPGA implementation was 20.48 s, while our GPU implementation took 5.93 s.

- An NVidia Tesla C1060 GPU with 240 processor cores operating at 1,296 GHz, with a total memory of 4 GB and a memory bandwidth of 102 GB/s. The cost of this platform is around US$1,000. The implementation was developed using the compute unified device architecture (CUDA) (http://www.nvidia.com/object/cuda_home.html), a collection of C extensions and a runtime library that allows general-purpose programming of NVidia GPUs.

Figure 6 shows the abundance maps for the most relevant endmembers extracted by the parallel implementations (which all provided exactly the same results as the serial version and the ENVI software). On the other hand, Table 2 shows the total time spent by our parallel implementations on the Thunderhead cluster and on the heterogeneous network. Two types of computation times were analyzed, sequential (those performed by the master node with no other parallel tasks active in the system) and parallel (the rest of computations, i.e., those performed by the master node and/or the workers in parallel). The latter includes the times in which the workers remain idle. In addition, Table 2 also displays the communication times, the total execution times, and the speedups (number of times that the parallel implementation was faster than the serial implementation). The total execution time measured for the serial version on a single Thunderhead processor was 1,163.05 s, while the same code executed on the fastest processor of the heterogeneous network was 1,098.63 s.

It can be seen from Table 2 that the times for sequential computations were always very low when compared to the times for parallel computations, which indicates high parallel efficiency of the developed implementations on both the cluster and the heterogeneous network, even for a high number of processors. The Beowulf cluster implementation took 4.98 s to produce the final output using 256 processors, achieving a speedup of 233.45, with regards to the serial implementation executed in one processor of the cluster. Interestingly, the speedup achieved in the heterogeneous network (16 CPUs) is similar to that achieved in the cluster with the same number of CPUs. In turn, the FPGA implementation was able to perform the same analysis in 20.48 s, using approximately 36% of the available hardware resources in the FPGA. It should be noted that this implementation is not fully optimized (better results can be achieved by increasing resource utilization on the FPGA). However, it is quite important to leave room in the FPGA for additional algorithms so that dynamic reconfiguration and algorithm selection can be performed on the fly. Finally, the GPU implementation took 5.93 s. This is close to real-time performance since the cross-track line scan time in AVIRIS, a push-broom instrument, is 8.3 ms (to collect 512 full pixel vectors). This introduces the need to process the considered scene (614×512 pixels) in less than 5.09 s to fully achieve real-time performance. Note that similar real-time performance could only be achieved in the Thunderhead cluster by resorting to 256 processors. Although clusters and networks are appealing for analyzing hyperspectral data sets already transmitted to Earth, they are very difficult to adapt to on-board data processing scenarios in which low-weight and low-power integrated components are essential to reduce mission payload. Despite the promising results achieved by the GPU implementation, the full incorporation of GPUs into satellite-based missions is still subject to further developments, particularly given the higher power consumption of GPUs when compared to FPGAs and the fact that radiation-hardened GPUs for space operation are not yet as widely available as their FPGA counterparts [12].

OUTLOOK

In this article, we have discussed the computational requirements involved in the analysis of hyperspectral images in the context of remote sensing applications. Through the analysis of a standard hyperspectral unmixing chain, we have illustrated different parallel systems and strategies to increase computational performance of hyperspectral imaging algorithms. Two of the considered techniques, i.e., cluster-based parallel computing and heterogeneous parallel computing, seem particularly appropriate for efficient information extraction from large hyperspectral data repositories containing data sets already transmitted to Earth. To fully address the real-time constraints introduced by many remote sensing applications, we have also discussed FPGA and GPU implementations of the hyperspectral unmixing chain, which are intended for on-board processing. A major goal is to overcome an existing limitation in many remote sensing and Earth observation systems: the bottleneck introduced by the bandwidth of the downlink connection from the observatory platform. Experimental results demonstrate that

our hardware implementations make appropriate use of computing resources in the considered FPGA and GPU architectures and further provide a response in (near) real time, which is believed to be acceptable in many remote sensing applications. The reconfigurability of FPGA systems on the one hand, and the low cost of GPU systems on the other, open many innovative perspectives from an application point of view. Although the experimental results presented for the considered unmixing chain are encouraging, further work is still needed to arrive at optimal parallel design and implementation of other more sophisticated hyperspectral processing algorithms, such as supervised classification or change detection.

AUTHORS

Antonio Plaza (aplaza@unex.es) is the head of the Hyperspectral Computing Laboratory at the University of Extremadura, Spain. He is an associate editor of *IEEE Transactions on Geoscience and Remote Sensing* and the coordinator of the Marie Curie Hyperspectral Imaging Network.

Javier Plaza (jplaza@unex.es) is an assistant professor with the Hyperspectral Computing Laboratory, University of Extremadura, Spain. His interests comprise hyperspectral image analysis and efficient implementations on clusters, FPGAs, and GPUs.

Abel Paz (apazgal@unex.es) is a staff member of Bull Spain working in the Center for Advanced Technologies of Extremadura and an assistant professor with the Hyperspectral Computing Laboratory, University of Extremadura, Spain. His interests comprise hyperspectral image analysis and efficient implementations on heterogeneous networks and GPUs.

Sergio Sánchez (sersanmar@unex.es) is a research associate with the Hyperspectral Computing Laboratory, University of Extremadura, Spain. His research interests comprise hyperspectral image analysis and efficient implementations on GPUs.

REFERENCES

[1] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for Earth remote sensing," *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.

[2] A. Plaza, J. A. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, J. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. 110–122, supplement 1, 2009, .

[3] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Processing Mag.*, vol. 19, no. 1, pp. 12–16, 2002.

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

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

[6] S. Kalluri, Z. Zhang, J. JaJa, S. Liang, and J. Townshend, "Characterizing land surface anisotropy from AVHRR data at a global scale using high performance computing," *Int. J. Remote Sens.*, vol. 22, no. 11, pp. 2171–2191, 2001.

[7] D. Heinz and C.-I. Chang, "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, 2000.

[8] 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, 2004.

[9] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of Aviris data," in *Proc. JPL Airborne Earth Science Workshop*, 1995, pp. 23–26.

[10] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proc. SPIE Image Spectrometry V*, 2003, vol. 3753, pp. 266–277.

[11] A. Lastovetsky and J. Dongarra, *High-performance Heterogeneous Computing*. New York: Wiley, 2009.

[12] E. El-Araby, T. El-Ghazawi, J. LeMoigne, and R. Irish, "Reconfigurable processing for satellite onboard automatic cloud cover assessment," *J. Real-Time Image Process.*, vol. 4, no. 3, pp. 245–259, 2009.

[SP]