Near real-time endmember extraction from remotely sensed hyperspectral data using NVidia GPUs

Sergio Sánchez[a], Gabriel Martín[a], Abel Paz[a], Antonio Plaza[a], and Javier Plaza[a]

[a]Hyperspectral Computing Laboratory
Department of Technology of Computers and Communications
University of Extremadura, Avda. de la Universidad s/n
E-10071 Cáceres, Spain

ABSTRACT

One of the most important techniques for hyperspectral data exploitation is spectral unmixing, which aims at characterizing mixed pixels. When the spatial resolution of the sensor is not fine enough to separate different spectral constituents, these can jointly occupy a single pixel and the resulting spectral measurement will be a composite of the individual pure spectra. The N-FINDR algorithm is one of the most widely used and successfully applied methods for automatically determining endmembers (pure spectral signatures) in hyperspectral image data without using a priori information. The identification of such pure signatures is highly beneficial in order to 'unmix' the hyperspectral scene, i.e. to perform sub-pixel analysis by estimating the fractional abundance of endmembers in mixed pixels collected by a hyperspectral imaging spectrometer. The N-FINDR algorithm attempts to automatically find the simplex of maximum volume that can be inscribed within the hyperspectral data set. Due to the intrinsic complexity of remotely sensed scenes and their ever-increasing spatial and spectral resolution, the efficiency of the endmember searching process conducted by N-FINDR depends not only on the size and dimensionality of the scene, but also on its complexity (directly related with the number of endmembers). In this paper, we develop a new parallel version of N-FINDR which is shown to scale better as the dimensionality and complexity of the hyperspectral scene to be processed increases. The parallel algorithm has been implemented on two different parallel systems, in which two different types of commodity graphics processing units (GPUs) from NVidia$^{\mathrm{TM}}$ are used to assist the CPU as co-processors. Commodity computing in GPUs is an exciting new development in remote sensing applications since these systems offer the possibility of (onboard) high performance computing at very low cost. Our experimental results, obtained in the framework of a mineral mapping application using hyperspectral data collected by the NASA Jet Propulsion Laboratory's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), reveal that the proposed parallel implementation compares favorably with the original version of N-FINDR not only in terms of computation time, but also in terms of the the accuracy of the solutions that it provides. The real-time processing capabilities of our GPU-based N-FINDR algorithms and other GPU algorithms for endmember extraction are also discussed.

Keywords: Hyperspectral imaging, endmember extraction, spectral unmixing, parallel N-FINDR, GPUs.

1. INTRODUCTION

Hyperspectral imaging[1,2] 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. The special characteristics of hyperspectral datasets pose different processing problems (especially due to the extremely high dimensionality of the data in the spectral domain, with hundreds of narrow spectral bands[3]), which must be necessarily tackled under specific mathematical formalisms. As a result, a wide range of techniques for hyperspectral image processing have been proposed in recent years.[4,5] One of such techniques is spectral unmixing, a very important task in remotely sensed hyperspectral data exploitation.[6] When the spatial resolution of the sensor is not fine enough to separate different spectral constituents,[7,8] these can jointly occupy a single pixel and the resulting spectral measurement will be a *mixed* pixel, i.e., a composite of the individual pure

Send correspondence to Antonio Plaza:
E-mail: aplaza@unex.es; Telephone: +34 927 257000 (Ext. 51662); URL: http://www.umbc.edu/rssipl/people/aplaza

Real-Time Image and Video Processing 2010, edited by Nasser Kehtarnavaz,
Matthias F. Carlsohn, Proc. of SPIE Vol. 7724, 772409 · © 2010 SPIE
CCC code: 0277-786X/10/$18 · doi: 10.1117/12.854365

spectra.[9] In order to define the mixture problem in mathematical terms, let us assume that a remotely sensed hyperspectral scene with n bands is denoted by $\mathbf{X}$, in which the pixel at the discrete, spatial coordinates (i, j) of the scene is represented by a feature vector given by $\mathbf{X}(i, j) = [x_1(i, j), x_2(i, j), \cdots, x_n(i, j)] \in \Re^n$, and $\Re$ denotes the set of real numbers corresponding to the pixel's spectral response $x_k(i, j)$ at sensor channels $k = 1, \ldots, n$. Under a linear mixture model assumption,[6] each pixel vector in the original scene can be modeled using the following expression:

$$\mathbf{X}(i, j) = \sum_{k=1}^{p} \Phi_k(i, j) \cdot \mathbf{E}_k + \mathbf{n}(i, j), \tag{1}$$

where $\mathbf{E}_k$ denotes the spectral response of the k-th endmember, $\Phi_z(i, j)$ is a scalar value designating the fractional abundance of the k-th at pixel $\mathbf{X}(i, j)$, p is the total number of endmembers, and $\mathbf{n}(i, j)$ is a noise vector. The solution of the linear spectral mixture problem described in (1) relies on the correct determination of a set of p endmembers denoted by $\{\mathbf{E}_k\}_{k=1}^{p}$.

Over the last decade, several algorithms have been developed for extraction of spectral endmembers directly from the input hyperspectral data set.[10] Winter's N-FINDR algorithm[11] is one of the most widely used and successfully applied methods for that purpose. This approach finds the set of pixels with the largest possible volume by "inflating" a simplex within the data. After reducing the dimensionality of the data from n to $p - 1$ (this is a feasible step, since typically $p << n$), a random set of p pixel vectors is initially selected from the input scene. In order to refine the initial estimate of endmembers, every pixel in the image must be evaluated in terms of pixel purity likelihood or nearly pure statehood. To achieve this, the volume is calculated for every pixel in the place of each endmember. A trial volume is calculated for every pixel in each endmember position by replacing that endmember and finding the volume. If the replacement results in a volume increase, the pixel replaces the endmember. This procedure is repeated until there are no more replacements of endmembers. While the endmember determination step of N-FINDR in the commercial version distributed by Pacific Spectral Technology* has been optimized for high speed processing, the computational performance of the algorithm depends on the accuracy of the initial random selection of endmembers and, most importantly, on the dimensions of the hyperspectral scene and the number of endmembers to be found, p.

Although commodity clusters have been used for speeding up computational performance of hyperspectral imaging applications in the past,[12,13] these systems are expensive and difficult to adapt to onboard data processing scenarios, in which low-weight and low-power integrated components are highly desirable to reduce mission payload.[14] In this regard, an exciting new development is the emergence of commodity graphics processing units (GPUs), which can now satisfy extremely high computational requirements, such as those introduced by hyperspectral imaging applications,[15–17] at very low cost. In this paper, we propose a new GPU-based implementation of the N-FINDR algorithm, which is used in this work as a representative case study of hyperspectral image processing algorithms. The remainder of the paper is organized as follows. Section 2 describes the original N-FINDR algorithm. Section 3 describes our parallel implementation. Section 4 quantitatively assesses our implementation in terms of both endmember extraction accuracy and parallel efficiency, using two different generations of commercial GPUs (8600GT and 8800GTX) from NVidia$^{\text{TM}}$, one of the most important GPU vendors†. Section 5 discusses real-time considerations for the proposed algorithm and for other similar parallel GPU algorithms in the area of hyperspectral image analysis. Finally, section 6 concludes the paper with some remarks and hints at plausible future research lines.

2. ORIGINAL N-FINDR ALGORITHM

The original N-FINDR algorithm developed by Winter[11] can be summarized by the following steps.[18]

1. *Feature reduction.* Apply a dimensionality reduction transformation such as the minimum noise fraction (MNF)[19] or the principal component analysis (PCA)[8] to reduce the dimensionality of the data from n to $p - 1$, where p is an input parameter to the algorithm (number of endmembers to be extracted).

*http://www.pacificspectral.com

†http://www.nvidia.com

2. *Initialization.* Let $\{\mathbf{E}_1^{(0)}, \mathbf{E}_2^{(0)}, \cdots, \mathbf{E}_p^{(0)}\}$ be a set of endmembers randomly extracted from the input data.

3. *Volume calculation.* At iteration $k \geq 0$, calculate the volume defined by the current set of endmembers as follows:

$$V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \cdots, \mathbf{E}_p^{(k)}) = \frac{\left| det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{E}_1^{(k)} & \mathbf{E}_2^{(k)} & \cdots & \mathbf{E}_p^{(k)} \end{bmatrix} \right|}{(p-1)!} \tag{2}$$

4. *Replacement.* For each pixel vector $\mathbf{X}(i,j)$ in the input hyperspectral data, recalculate the volume by testing the pixel in all p endmember positions, i.e., first calculate $V(\mathbf{X}(i,j), \mathbf{E}_2^{(k)}, \cdots, \mathbf{E}_p^{(k)})$, then calculate $V(\mathbf{E}_1^{(k)}, \mathbf{X}(i,j), \cdots, \mathbf{E}_p^{(k)})$, and so on, until $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \cdots, \mathbf{X}(i,j))$. If none of the p recalculated volumes is greater than $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \cdots, \mathbf{E}_p^{(k)})$, then no endmember is replaced. Otherwise, the combination with maximum volume is retained. Let us assume that the endmember absent in the combination resulting in the maximum volume is denoted by $\mathbf{E}_j^{(k+1)}$. In this case, a new set of endmembers is produced by letting $\mathbf{E}_j^{(k+1)} = \mathbf{X}(i,j)$ and $\mathbf{E}_i^{(k+1)} = \mathbf{E}_i^{(k)}$ for $i \neq j$. The replacement step is repeated in iterative fashion, using as many iterations as needed until there are no more replacements of endmembers.

3. PARALLEL IMPLEMENTATION OF THE N-FINDR ALGORITHM

Our parallel implementation of N-FINDR can be summarized by the following steps. It should be noted that step 1 (*feature reduction*) of the original N-FINDR algorithm has not been implemented in parallel.

1. *Initialization.* Let $\{\mathbf{E}_1^{(0)}, \mathbf{E}_2^{(0)}, \cdots, \mathbf{E}_p^{(0)}\}$ be a set of endmembers randomly extracted from the input data. The volume defined by this set of endmembers, $V(\mathbf{E}_1^{(0)}, \mathbf{E}_2^{(0)}, \cdots, \mathbf{E}_p^{(0)})$ is calculated using Eq. (2). Due to the low computational complexity of this step, it is not implemented in parallel.

2. *Data partitioning.* Assign a different pixel vector to each processor of the parallel system in a pre-defined row-column order, i.e., from the first pixel in the first image row, $\mathbf{X}(1,1)$, to the last pixel in the first image row, $\mathbf{X}(1,C)$; then, from the first pixel in the second image row, $\mathbf{X}(2,1)$, to the last pixel in the second image row, $\mathbf{X}(2,C)$; and so on, until the last pixel in the last image row, $\mathbf{X}(R,C)$, is processed. Here, R denotes the total number of rows and C denotes the total number of columns.

3. *Replacement.* At iteration $k \geq 0$, recalculate the volume by first testing a block of R pixel vectors (in parallel) in all p endmember positions. This situation is illustrated graphically in Fig. 1. Let us assume that one of such pixels, say, the one with spatial coordinates (i,j), is allocated to processor t, with $t \in \{1, \cdots, T\}$, being T the total number of processors in the parallel system. If we denote such pixel as $\mathbf{X}_t(i,j)$, then p volumes can be calculated at processor t by testing $\mathbf{X}_t(i,j)$ in all p endmember positions, i.e., processor t first calculates $V(\mathbf{X}_t(i,j), \mathbf{E}_2^{(k)}, \cdots, \mathbf{E}_p^{(k)})$, then $V(\mathbf{E}_1^{(k)}, \mathbf{X}_t(i,j), \cdots, \mathbf{E}_p^{(k)})$, and so on, until $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \cdots, \mathbf{X}_t(i,j))$. If none of the p recalculated volumes at processor t is greater than $V(\mathbf{E}_1^{(k)}, \mathbf{E}_2^{(k)}, \cdots, \mathbf{E}_p^{(k)})$, then no endmember is replaced. Otherwise, the combination with maximum volume is retained along with the coordinates of the pixel and the endmember that was replaced at each processor (see Fig. 1). If we denote the endmember absent in the combination resulting in the maximum volume by $\mathbf{E}_j^{(k+1)}$, then a new set of endmembers at processor t is produced at processor t by letting $\mathbf{E}_j^{(k+1)} = \mathbf{X}_t(i,j)$ and $\mathbf{E}_i^{(k+1)} = \mathbf{E}_i^{(k)}$ for $i \neq j$. After R pixels have been processed in parallel, the maximum volume resulting from all iterations is selected and replaced, and the procedure continues until all the columns in the original hyperspectral image have been exhausted. The replacement step is repeated in iterative fashion, using as many iterations as needed until there are no more replacements of endmembers.

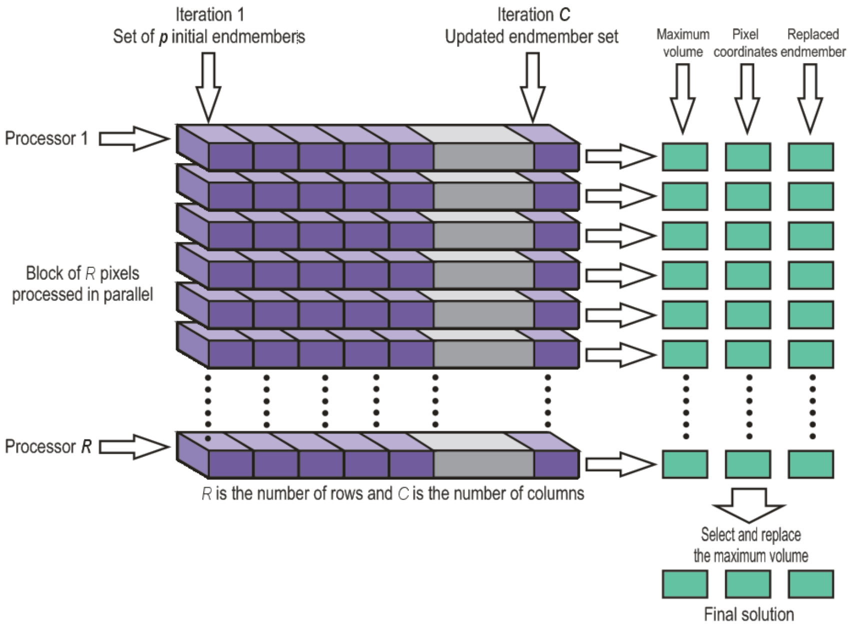

Iteration 1
Set of p initial endmembers
Iteration C
Updated endmember set
Maximum volume
Pixel coordinates
Replaced endmember
Processor 1
Block of R pixels processed in parallel
Processor R
R is the number of rows and C is the number of columns
Select and replace the maximum volume
Final solution

Figure 1. Diagram illustrating the proposed parallel implementation of N-FINDR algorithm.

4. EXPERIMENTAL RESULTS

The parallel implementation of N-FINDR has been validated with regards to a serial version of the original N-FINDR, using a well-known hyperspectral scenes collected by the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS)[20] over the Cuprite mining district in Nevada[‡]. This scene has been widely used to validate the performance of endmember extraction algorithms.[21, 22] The portion used in experiments corresponds to a 250×191-pixel subset of the sector labeled as f970619t01p02_r02_sc03.a.rfl in the online data. Fig. 2(a) shows the spectral band at 667.3 nm wavelength of the considered scene, which comprises 224 spectral bands between 0.4 and 2.5 μm and nominal 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 signal-to-noise ratio (SNR) in those bands, leaving a total of 188 spectral bands and an image size of 17.4 MB. A library of reference spectral signatures collected by U.S. Geological Survey (USGS) is available for the Cuprite scene[§]. A few selected spectra from the USGS library, corresponding to minerals: alunite, buddingtonite, calcite, kaolinite and muscovite [see Fig. 2(b)], are used in this work to substantiate endmember signature purity.

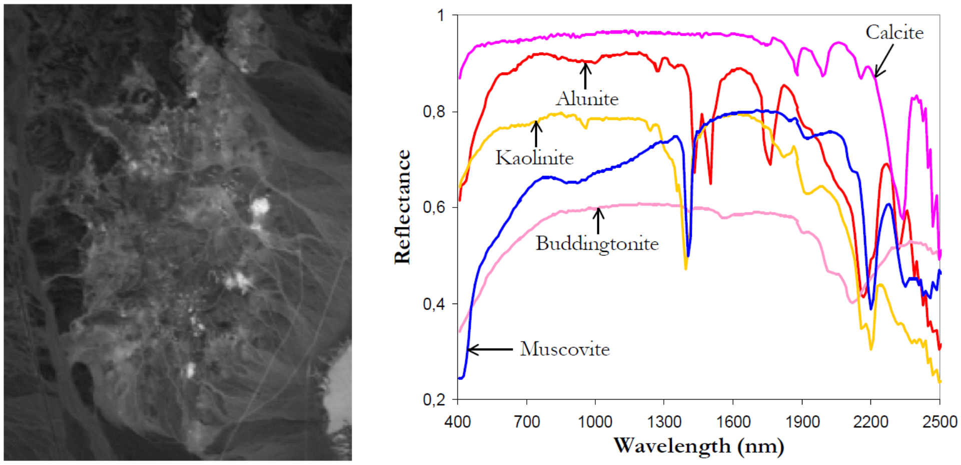
It should be noted that, in our experiments, the *feature reduction* step of N-FINDR has been conducted using the MNF transformation. The *initialization* step has been performed so that both implementations (serial and parallel) are run 50 times and receive exactly the same 50 sets of randomly chosen endmember pixels to start the process, and the volume-based scores provided by each implementation after the 50 runs are reported. It is worth noting that the estimation of the number of endmembers, p, to be extracted by each implementation has been conducted using the virtual dimensionality (VD) concept.[23] This approach uses a Neyman-Pearson detector as a decision maker based on a prescribed false alarm probability, P_F. According to recent studies,[24] a reasonable empirical choice is $P_F = 10^{-4}$, which resulted in an estimate of $p = 16$ for the considered AVIRIS scene.

4.1 Experiment 1: Analysis of volume estimations

Fig. 3(a) analyzes the volume of the simplex provided for the considered implementations in each of the different 50 runs. Each bar in Fig. 3(a) reports the percentage of volume estimated by each method out of the total

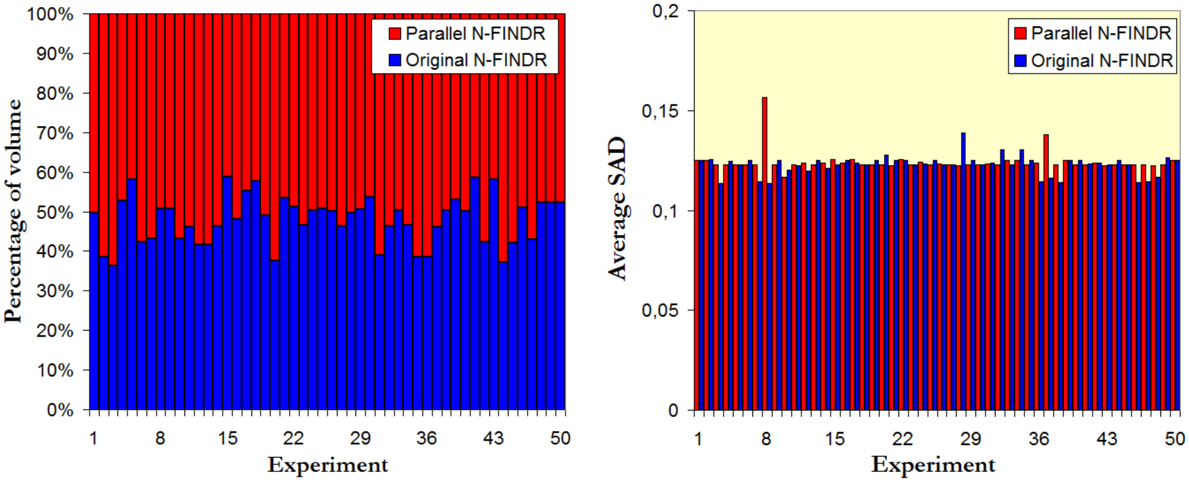
[‡] http://aviris.jpl.nasa.gov/html/aviris.freedata.html

[§] http://speclab.cr.usgs.gov



(a) (b)

Figure 2. (a) Spectral band at 667.3 nm wavelength of a portion of the AVIRIS Cuprite scene. (b) USGS spectral signatures of five representative minerals in the Cuprite mining district.



(a) (b)

Figure 3. (a) Comparison of volume estimates, and (b) comparison of average SAD with regards to five reference USGS mineral signatures obtained for the original and parallel N-FINDR implementations.

cumulative volume estimated by both methods. Therefore, the optimal situation in this experiment would be achieved when the 50 bars (each of which corresponds to a different experiment) are made up of 50% volume for the original N-FINDR and 50% volume for the parallel implementation, meaning that the two implementations reach exactly the same volume in the final endmember solution. As shown by Fig. 3(a), the two compared methods achieve comparatively very similar volume estimations. Out of 50 experiments, the parallel version produced higher volume than the serial version in 25 of them.

4.2 Experiment 2: Analysis of extracted endmembers

Our second experiment analyzes the spectral purity of the endmembers extracted by the two considered implementations. This is assessed by reporting the average spectral angle distance (SAD) scores obtained after comparing the USGS library signatures of the main five minerals present in the Cuprite scene (*alunite*, *buddingtonite*, *calcite*, *kaolinite* and *muscovite*) with the corresponding endmembers extracted by the different N-FINDR implementations, where each library signature was matched to one of the endmembers extracted by a certain algorithm in terms of the smaller SAD value observed across the full endmember set. Fig. 3(b) reports the average SAD values obtained by each method in each of the different 50 runs. As shown by Fig. 3(b), both the serial

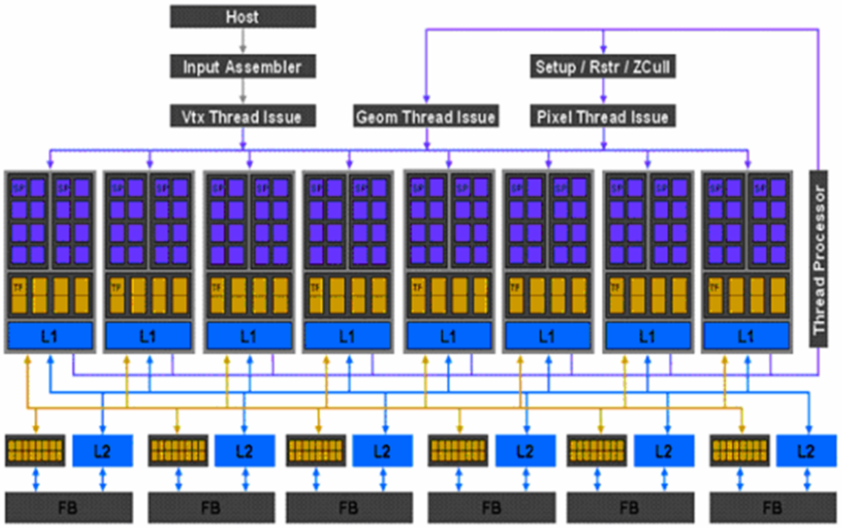


Figure 4. NVidia^{TM} GeForce 8800GTX architecture.

and parallel implementations achieve comparatively similar average SAD values, indicating that their capacity to extract endmembers which are similar (in spectral angle sense) with regards to five highly representative USGS mineral signatures is comparable, even though the original N-FINDR implementation is slightly better (lower average SAD scores). Out of 50 experiments, the parallel version produced lower average SAD score than the serial version in 27 of them.

4.3 Experiment 3: Analysis of parallel performance in different GPU architectures

The proposed endmember extraction algorithm has been implemented on two different generations of NVidia^{TM} GPUs. The parallel algorithm was implemented using NVidia^{TM} CUDA, a collection of C extensions and a runtime library. CUDAs functionality primarily allows a developer to write C functions to be executed on the GPU. CUDA also includes memory management and execution configuration, so that a developer can control the number of GPU processors and threads that are to be invoked during a function's execution. The CPU implementations were developed using the Intel C/C++ compiler. Two different systems were used in our experiments. The first one is based on an Intel Quad-Core CPU running at 2.4 GHz and with 2 GB of RAM. This computer is equipped with an NVidia^{TM} GeForce 8600GT with 4 multiprocessors and 255 MB of global memory. The second system used in experiments is based on an Intel Core 2 Duo CPU running at 2.33 GHz and with 2 GB of RAM. The computer is equipped with an NVidia^{TM} GeForce 8800GTX with 16 multiprocessors, each composed of 8 SIMD processors operating at 1350 Mhz. Each multiprocessor has 8192 registers, a 16 KB parallel data cache of fast shared memory, and access to 768 MB of global memory. The GPU architecture is graphically illustrated in Fig. 4.

Fig. 5 shows the execution times measured for the CPU and GPU-based implementations, respectively, for each of the 50 experiments conducted in the two considered systems, with Fig. 5(a) displaying the timing results in the system with the 8600GT GPU and Fig. 5(b) displaying the timing results in the system with the 8800GTX GPU. All the cores available in the CPU systems were used when the timing the serial implementation. The C function *clock()* was used for timing the CPU implementation and the CUDA timer was used for the GPU implementation. The time measurement was started right after the hyperspectral image file was read to the CPU memory and stopped right after the results of the N-FINDR algorithm were obtained and stored in the CPU memory. After calculating the mean scores from the different times reported in Fig. 5, it was observed that the serial version of N-FINDR took 68320 ms (on average) to process the considered AVIRIS scene in the system with the 8600GT GPU, while the same serial version took 68660 ms (on average) to process the same scene in the system with the 8800GTX GPU. In contrast, the parallel version of the algorithm took 39100 ms (on average) to process the considered AVIRIS scene in the system with the 8600GT GPU, while the same parallel version took 26385 ms (on average) to process the considered AVIRIS scene in the system with the 8800GTX

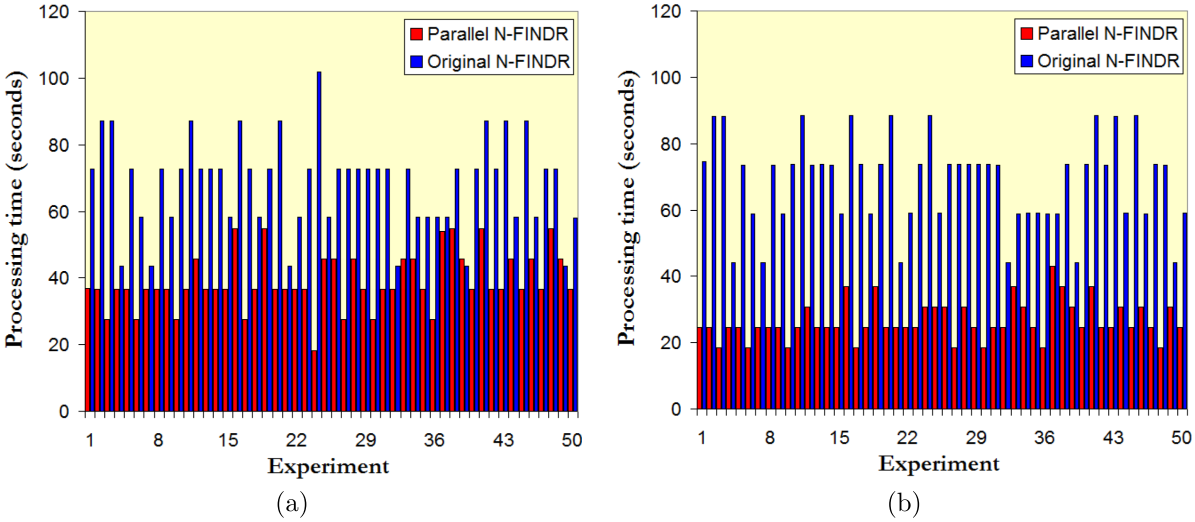


Figure 5. (a) Processing times in the NVidia$^{\text{TM}}$ 8600GT GPU. (b) Processing times in the NVidia$^{\text{TM}}$ 8800GTX GPU.

GPU. It should be noted that the serial processing times reported in Fig. 5 correspond to versions of N-FINDR which have been carefully optimized in the two considered systems. For reference, we also measured the times achieved by the serial version of our parallel implementation, obtaining very similar average processing times (62192 ms and 67696 ms, respectively) to those reported for the optimized serial versions executed in the two considered systems.

Although the proposed implementation can still be optimized, we emphasize that (according to our experiments with other scenes with different sizes and number of endmembers), the speedup achieved by the proposed parallel version with regards to the serial version depends on the value of p (number of endmembers). In other words, for large values of p and large images, the proposed parallel algorithm can achieve better speedups than for smaller values of p and smaller images. We believe that this is an interesting and highly desirable feature for the proposed parallel algorithm, mainly because the size, spectral dimensionality and spatial resolution of hyperspectral images is ever improving resulting from technological advances in sensor design and instrumentation.

5. REAL-TIME CONSIDERATIONS

Before concluding the paper, it is important to emphasize that the processing times reported in the previous section are not in real-time since the cross-track line scan time in AVIRIS, a push-broom instrument, is quite fast (8.3 msec to collect 512 full pixel vectors). This introduces the need to process the considered scene (250 × 191 pixels) in approximately 3.68 seconds to fully achieve real-time performance. Despite the fact that the current implementation of N-FINDR could only achieve processing times around 20 seconds for the considered scene, we believe that further optimizations for N-FINDR are possible in future developments by making a better use of GPU computing resources and reducing the granularity of the problem. In this regard, recent experiments with an optimized version of N-FINDR in an NVidia Tesla GPU architecture resulted in speedups on the order of 10 times faster (instead of 3 times faster reported in this paper) when comparing the GPU version with regards to an optimized serial version of the same code executed in the same machine without resorting to the CPU. In addition to this, we have recently developed a GPU version of a different endmember extraction algorithm called orthogonal subspace projection (OSP),[25,26] with similar performance to N-FINDR in terms of endmember extraction accuracy and better parallel performance.[27] Specifically, the implementation of OSP using an NVidia GeForce GTX 275 GPU was able to process a 614 × 512 hyperspectral image in 5.51 seconds as compared to an execution of the same algorithm in a desktop PC with 4 cores (all of them were used to implement the reference version) which took 204.09 seconds. This means that the GPU version of OSP was 37 times faster than the multi-core implementation of the same algorithm, thus achieving near real-time performance since the limit to process a hyperspectral scene with 614 × 512 pixels is 5.09 seconds. Another strategy that we have in mind to further speedup the performance of N-FINDR in the task of endmember extraction is to apply a fast spectral pre-processing operation to remove those pixels which are not sufficiently pure prior to the endmember searching process implemented by N-FINDR. In this regard, we are currently experimenting with NVidia Tesla architectures in order to fully achieve the goal of real-time endmember extraction from hyperspectral imagery, which may allow incorporation of processing hardware onboard hyperspectral imaging instruments for real-time data exploitation at the same time as the data is collected by the sensor.

6. CONCLUSIONS AND FUTURE RESEARCH LINES

The ever increasing spatial and spectral resolutions that will be available in the new generation of hyperspectral instruments for remote observation of the Earth anticipates significant improvements in the capacity of these instruments to uncover spectral signals in complex real-world analysis scenarios. Such capacity demands parallel processing techniques which can cope with the requirements of time-critical applications and properly scale with image size, dimensionality and complexity. In order to address such need, in this paper we have developed a new parallel version of the popular N-FINDR algorithm for automatic endmember extraction in the context of spectral unmixing applications. The performance of the proposed parallel algorithm has been evaluated (in terms of the quality of the solutions it provides and its parallel performance) on two different GPU generations of commercial GPUs (8600GT and 8800GTX) from NVidia^{TM}. Our experimental results indicate that the accuracy of the parallel algorithm is very similar with regards to the original N-FINDR. Also, the parallel algorithm performs better with latest-generation GPUs, thus taking advantage of the increased processing power of such units. Most importantly, the algorithm scales better as the size and complexity (given by the number of endmembers, p) of the hyperspectral image to be processed is increased. Resulting from these observations, we conclude that our parallel algorithm may cope well with the requirements introduced by the new generation of hyperspectral instruments. Although the proposed implementation is not strictly in real-time, we expect to achieve this consideration in future developments as it is already the case with other endmember extraction algorithms that we implemented in similar GPU architectures (NVidia Tesla and GeForce GTX 275) which are very close to perform in real-time mode. In order to fully substantiate the aforementioned remarks, further experimentation with additional hyperspectral scenes and GPU architectures is highly desirable.

7. ACKNOWLEDGEMENT

This work has been supported by the European Community's Marie Curie Research Training Networks Programme under reference MRTN-CT-2006-035927, Hyperspectral Imaging Network (HYPER-I-NET). This work has also been supported by the Spanish Ministry of Science and Innovation (HYPERCOMP/EODIX project, reference AYA2008-05965-C04-02). Sergio Sánchez and Gabriel Martín are sponsored by research fellowships with references PTA2009-2611-P and BES-2009-017737, respectively, both associated to the aforementioned project.

REFERENCES

1. A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for Earth remote sensing," *Science* **228**, pp. 1147–1153, 1985.
2. C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Kluwer Academic/Plenum Publishers: New York, 2003.
3. D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*, John Wiley & Sons: New York, 2003.
4. 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 Sensing of Environment* **113**, pp. 110–122, 2009.
5. M. E. Schaepman, S. L. Ustin, A. Plaza, T. H. Painter, J. Verrelst, and S. Liang, "Earth system science related imaging spectroscopy - an assessment," *Remote Sensing of Environment* **113**, pp. 123–137, 2009.
6. J. B. Adams, M. O. Smith, and P. E. Johnson, "Spectral mixture modeling: a new analysis of rock and soil types at the Viking Lander 1 site," *Journal of Geophysical Research* **91**, pp. 8098–8112, 1986.
7. J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, Springer, 2006.

8. R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing, 2nd ed.*, Academic Press: NY, 1997.
9. N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine* **19**(1), pp. 44–57, 2002.
10. A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing* **42**(3), pp. 650–663, 2004.
11. M. Winter, "N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proceedings of SPIE*, **3753**, pp. 266–270, 1999.
12. A. Plaza, D. Valencia, J. Plaza, and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral imagery," *Journal of Parallel and Distributed Computing* **66**, pp. 345–358, 2006.
13. A. Plaza and C.-I. Chang, *High Performance Computing in Remote Sensing*, Taylor & Francis: Boca Raton, FL, 2007.
14. A. Plaza, "Preface to the special issue on architectures and techniques for real-time processing of remotely sensed images," *Journal of Real-Time Image Processing* **4**, pp. 191–193, 2009.
15. J. Setoain, M. Prieto, C. Tenllado, A. Plaza, and F. Tirado, "Parallel morphological endmember extraction using commodity graphics hardware," *IEEE Geoscience and Remote Sensing Letters* **43**, pp. 441–445, 2007.
16. A. Plaza, J. Plaza, S. Sanchez, and A. Paz, "Lossy hyperspectral image compression tuned for spectral mixture analysis applications on nvidia graphics processing units," in *Proceedings of SPIE*, **7455, 74550F**, 2009.
17. Y. Tarabalka, T. V. Haavardsholm, I. Kasen, and T. Skauli, "Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and gpu processing," *Journal of Real-Time Image Processing* **4**, pp. 287–300, 2009.
18. M. Zortea and A. Plaza, "A quantitative and comparative analysis of different implementations of N-FINDR: A fast endmember extraction algorithm," *IEEE Geoscience and Remote Sensing Letters* **6**, pp. 787–791, 2009.
19. A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Transactions on Geoscience and Remote Sensing* **26**, pp. 65–74, 1988.
20. R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment* **65**(3), pp. 227–248, 1998.
21. R. N. Clark, G. A. Swayze, K. E. Livo, R. F. Kokaly, S. J. Sutley, J. B. Dalton, R. R. McDougal, and C. A. Gent, "Imaging spectroscopy: Earth and planetary remote sensing with the usgs tetracorder and expert systems," *Journal of Geophysical Research* **108**, pp. 1–44, 2003.
22. G. Swayze, R. N. Clark, F. Kruse, S. Sutley, and A. Gallagher, "Ground-truthing AVIRIS mineral mapping at Cuprite, Nevada," *Proc. JPL Airborne Earth Sci. Workshop* , pp. 47–49, 1992.
23. C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing* **42**(3), pp. 608–619, 2004.
24. A. Plaza and C.-I. Chang, "Impact of initialization on design of endmember extraction algorithms," *IEEE Transactions on Geoscience and Remote Sensing* **44**(11), pp. 3397–3407, 2006.
25. J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection," *IEEE Transactions on Geoscience and Remote Sensing* **32**(4), pp. 779–785.
26. H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Transactions on Aerospace and Electronic Systems* **39**(4), pp. 1232–1249, 2003.
27. A. Paz, A. Plaza, and J. Plaza, "Comparative analysis of different implementations of a parallel algorithm for automatic target detection and classification of hyperspectral images," in *Proceedings of SPIE*, **7455, 74550X**, 2009.