PARALLEL IMPLEMENTATION OF THE N-FINDR ENDMEMBER EXTRACTION ALGORITHM ON COMMODITY GRAPHICS PROCESSING UNITS

Sergio Sánchez, Gabriel Martín and Antonio Plaza

Department of Technology of Computers and Communications
University of Extremadura, Avda. de la Universidad s/n, E-10071 Caceres, Spain
Phone: +34 927 257000 (Ext. 51662) - Fax: +34 927 257203
E-mail: {sersanmar, gamahefpi, aplaza}@unex.es

1. INTRODUCTION

A wide range of techniques for hyperspectral image processing have been proposed in recent years [1, 2]. One of such techniques is spectral unmixing, a very important task in remotely sensed hyperspectral data exploitation [3]. 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 *mixed* pixel, i.e., a composite of the individual pure spectra [4]. 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 [3], 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 [5]. Winter's N-FINDR algorithm [6] 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[1] 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 [7], 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. An exciting new development is the emergence of commodity graphics processing units (GPUs), which can now satisfy extremely high computational requirements at low cost [8]. In this paper, we propose a new GPU-based implementation of the N-FINDR algorithm. The proposed implementation is quantitatively assessed in terms of both endmember extraction accuracy and parallel efficiency, using two different generations of commercial GPUs (8600GT and 8800GTX) from NVidia$^{\text{TM}}$, a famous GPU vendor[2].

[1] http://www.pacificspectral.com

[2] http://www.nvidia.com

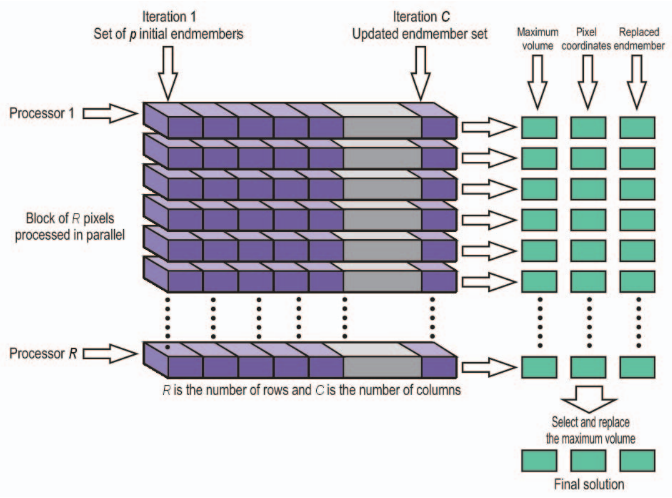

Iteration 1
Set of p initial endmembers
Iteration C
Updated endmember set
Maximum volume
Pixel coordinates
Replaced endmember
Processor 1
Block of R pixels processed in parallel
Processor R
R is the number of rows and C is the number of columns
Select and replace the maximum volume
Final solution

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

2. PARALLEL IMPLEMENTATION OF THE N-FINDR ALGORITHM

1. *Feature reduction*. Apply a dimensionality reduction transformation such as the minimum noise fraction (MNF) [9] or the principal component analysis (PCA) [10] 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). This step is not implemented in parallel.

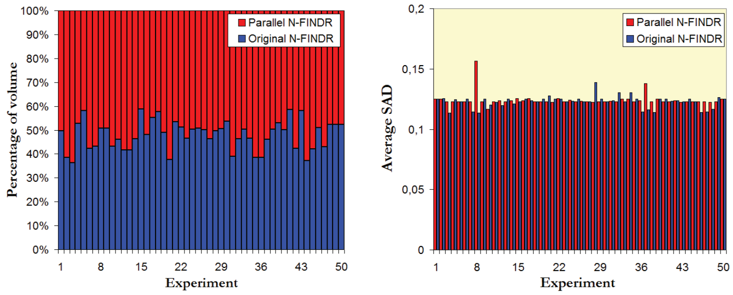
2. *Initialization*. Let $\{\mathbf{E}_1^{(0)}, \mathbf{E}_2^{(0)}, \cdots, \mathbf{E}_p^{(0)}\}$ be a set of endmembers randomly extracted from the input data. The volume defined by this set of endmembers, $V(\mathbf{E}_1^{(0)}, \mathbf{E}_2^{(0)}, \cdots, \mathbf{E}_p^{(0)})$ is then calculated. Due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

3. *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.

4. *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.

3. 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) over the Cuprite



(a) (b)

Fig. 2. (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.

mining district in Nevada[3]. This scene has been widely used to validate the performance of endmember extraction algorithms. 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. 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[4]. A few selected spectra from the USGS library, corresponding to minerals: *alunite*, *buddingtonite*, *calcite*, *kaolinite* and *muscovite* are used in this work to substantiate endmember signature purity. 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 [11], which resulted in an estimate of $p = 16$ for the considered AVIRIS scene.

3.1. Experiment 1: Analysis of volume estimations

Fig. 2(a) analyzes the volume of the simplex provided for the considered implementations in each of 50 different runs. Each bar in Fig. 2(a) reports the percentage of volume estimated by each method out of the total 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. 2(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.

3.2. Experiment 2: Analysis of extracted endmembers

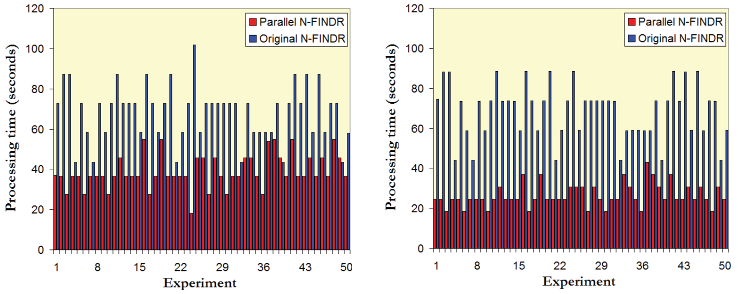
Our second experiment analyzes the spectral purity of the endmembers extracted by the two considered implementations. This is assessed by reporting the average spectral angle distance (SAD) scores obtained after comparing the USGS library signatures of the main five minerals present in the Cuprite scene (*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. 2(b) reports the average SAD values obtained by each method in each of the different 50 runs. As shown by Fig. 2(b), both the serial 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 one in 27 of them.

3.3. Experiment 3: Analysis of parallel performance in different GPU architectures

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$^{\text{TM}}$ 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

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

[4] http://speclab.cr.usgs.gov



(a) (b)

Fig. 3. (a) Processing times in the NVidia$^{\text{TM}}$ 8600GT GPU. (b) Processing times in the NVidia$^{\text{TM}}$ 8800GTX GPU.

with 2 GB of RAM. The computer is equipped with an NVidia$^{\text{TM}}$ 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. Fig. 3 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. 3(a) displaying the timing results in the system with the 8600GT GPU and Fig. 3(b) displaying the timing results in the system with the 8800GTX GPU. It should be noted that the serial processing times reported in Fig. 3 correspond to versions of N-FINDR which have been carefully optimized in the two considered systems. 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. Further experimentation with additional hyperspectral scenes and GPU architectures is desirable.

4. REFERENCES

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

[2] 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*, vol. 113, pp. 123–137, 2009.

[3] 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*, vol. 91, pp. 8098–8112, 1986.

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

[5] 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.

[6] M.E. Winter, "N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proceedings of SPIE*, 1999, vol. 3753, pp. 266–270.

[7] A. Plaza, D. Valencia, J. Plaza, and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral imagery," *Journal of Parallel and Distributed Computing*, vol. 66, pp. 345–358, 2006.

[8] J. Setoain, M. Prieto, C. Tenllado, A. Plaza, and F. Tirado, "Parallel morphological endmember extraction using commodity graphics hardware," *IEEE Geosci. Remote Sens. Lett.*, vol. 43, pp. 441–445, 2007.

[9] 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 Trans. Geosci. Remote Sens.*, vol. 26, pp. 65–74, 1988.

[10] R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing, 2nd ed.*, Academic Press: NY, 1997.

[11] 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, 2004.