# PARALLEL IMPLEMENTATION OF ENDMEMBER EXTRACTION ALGORITHMS USING NVIDIA GRAPHICAL PROCESSING UNITS

*Antonio Plaza, Javier Plaza, Sergio Sánchez*

Department of Technology of Computers and Communications
University of Extremadura, Avda. de la Universidad s/n, E-10071 Caceres, Spain
Phone: +34 927 257195 – Fax: +34 927 257203
Contact e-mail: aplaza@unex.es

## 1. INTRODUCTION

Spectral mixture analysis has been an alluring exploitation goal since the earliest days of hyperspectral imaging [1]. No matter the spatial resolution, in natural environments the spectral signature for a nominal pixel is invariably a mixture of the signatures of the various materials found within the spatial extent of the ground instantaneous field view of the sensor. In hyperspectral imagery, the number of spectral bands usually exceeds the number of pure spectral components, called *endmembers* in hyperspectral analysis terminology, and the unmixing problem is cast in terms of an over-determined system of equations in which, given the correct set of endmembers allows determination of the actual endmember abundance fractions through a numerical inversion process. Since each observed spectral signal is the result of an actual mixing process, the driving abundances must obey two constraints. First, all abundances must be non-negative. Second, the sum of abundances for a given pixel must be unity. However, it is the derivation and validation of the correct suite of endmembers that has remained a challenging and goal for the past years.

Over the last decade, several algorithms have been developed for automatic extraction of endmembers from hyperspectral data sets [1]. A popular approach focused on exploiting the spectral information in the data has been the pixel purity index (PPI) [2]. Other algorithms, such as the automatic morphological endmember extraction (AMEE) [3], integrate the spatial and the spectral information in the data. While hyperspectral imaging algorithms such as PPI or AMEE hold great promise for Earth science image analysis, they also introduce new processing challenges, in particular, for very high-dimensional data sets. From a computational perspective, these algorithms exhibit inherent parallelism at multiple levels: across pixel vectors (coarse grained pixel-level parallelism), across spectral information (fine grained spectral-level parallelism), and even across tasks (task-level parallelism). As a result, they map nicely to massively parallel systems made up of commodity CPUs (e.g., Beowulf clusters) [4]. Unfortunately, these systems are expensive and difficult to adapt to onboard remote sensing data processing scenarios, in which low-weight and low-power integrated components are mandatory to reduce mission payload.

An exciting recent development in the field of commodity computing is the emergence of programmable graphics processing units (GPUs) [5], which bear many features of vector processors. The speed of graphics hardware doubles approximately every six months, which is much faster than the improving rate of the CPU. Currently, state-of-the-art GPUs deliver peak performances in the order of 300 Gflops, which is more than one order of magnitude over high-end microprocessors. The ever-growing computational requirements introduced by hyperspectral imaging applications can fully benefit from this hardware and take advantage of the compact size and relatively low cost of these units, which make them appealing for onboard data processing at much lower costs than those introduced by other hardware devices [5].

In this paper, we develop three new GPU-based implementations of endmember extraction algorithms: the PPI, a kernel version of the PPI which represents a novel contribution of this work, and the spatial-spectral AMEE algorithm. We also provide a GPU-based implementation of the fully constrained linear spectral unmixing algorithm. The algorithms have been implemented on an NVidia GeForce 8800 GTX GPU[1], using the *compute unified device architecture* (CUDA)[2], a standard in the development of general-purpose implementations for NVidia GPUs.

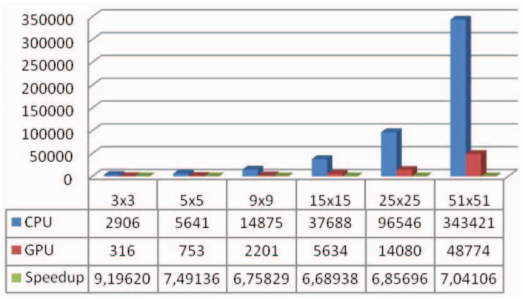
[1] http://www.nvidia.com/page/geforce_8800.html
[2] http://www.nvidia.com/object/cuda_home.html

## 2. ALGORITHMS AND IMPLEMENTATION

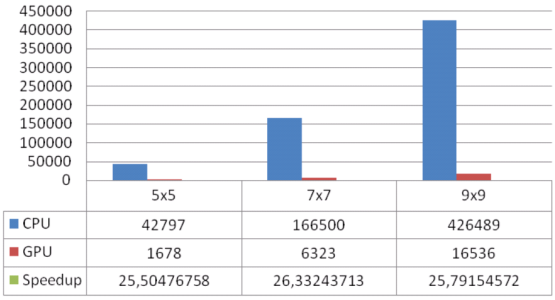
The three algorithms considered in this work are the PPI algorithm, a kernel version of the PPI (called KPPI), and the AMEE algorithm. The PPI proceeds by generating a large number of random, N-dimensional unit vectors called "skewers" through the dataset [3]. Every data point is projected onto each skewer, and the data points that correspond to extrema in the direction of a skewer are tallied. The pixels with the highest tallies are considered the final endmembers. On the other hand, the AMEE begins by searching spatial neighborhoods around each pixel in the image for the most spectrally pure and mostly highly mixed pixel. This task is performed by using extended mathematical morphology operators of dilation and erosion, respectively [4]. Each spectrally pure pixel is assigned an "eccentricity" value, which is calculated as the spectral angle distance between the most spectrally pure and mostly highly mixed pixel for the given spatial neighborhood. This process is repeated iteratively for larger spatial neighborhoods up to a maximum size that is pre-determined. The final endmember set is obtained by applying a threshold to the resulting "eccentricity" image. Finally, the KPPI can be seen as a hybrid between the PPI and the AMEE. It begins by searching spatial neighborhoods around each pixel in the image, but instead of using morphological operations of erosion and dilation, it looks for the most spectrally pure pixels in the local neighborhood by applying a variant of the original PPI in each local neighborhood. The three algorithms above have been implemented using CUDA kernels (a set of abstractions that allow performing modular tasks in the GPU). The first kernel (*stream uploading*) is common to the three methods. For the PPI and KPPI, three additional kernels are needed: *skewer generation*, *extreme projections* and *maximum/minimum*. Finally, for the AMEE five additional kernels are needed: *spectral angle distance*, *cumulative distance*, *maximum/minimum vector calculation*, *erosion/dilation*, and *eccentricity calculation*.

## 3. EXPERIMENTAL RESULTS

The hyperspectral data set used in our experiments is the well-known AVIRIS Cuprite scene[3]. The reflectance spectra of ten U.S. Geological Survey (USGS) ground mineral spectra were used as ground-truth spectra, revealing that the three considered algorithms were able to extract very good candidate endmembers from the input scene. In order to study the scalability of our CPU and GPU-based implementations, we tested them on the full scene (512 samples by 614 lines and 191 spectral bands after removal of low SNR channels, for a total size of 120 MB). The CPU used in experiments was an Intel Core 2 Quad Q6600 processor at 2.4 GHz and with 4 GB of RAM. Figure 1 shows the processing times (in milliseconds) and the speedups achieved over the CPU for the KPPI (left) and AMEE (right) using different window sizes. The speedups obtained make the GPU processing comparable to the data rate of the AVIRIS sensor, thus enabling near real-time endmember extraction.




| | 3x3 | 5x5 | 9x9 | 15x15 | 25x25 | 51x51 |
|---|---|---|---|---|---|---|
| CPU | 2906 | 5641 | 14875 | 37688 | 96546 | 343421 |
| GPU | 316 | 753 | 2201 | 5634 | 14080 | 48774 |
| Speedup | 9,19620 | 7,49136 | 6,75829 | 6,68938 | 6,85696 | 7,04106 |




| | 5x5 | 7x7 | 9x9 |
|---|---|---|---|
| CPU | 42797 | 166500 | 426489 |
| GPU | 1678 | 6323 | 16536 |
| Speedup | 25,50476758 | 26,33243713 | 25,79154572 |

Figure 1. Processing times (milliseconds) and speedups measured for the GPU versions of the KPPI (left) and AMEE (right) using different window sizes.

## 4. REFERENCES


[1] 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.

[2] J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping Target Signatures Via Partial Unmixing of Aviris Data," *Proc. JPL Airborne Earth Sci. Workshop*, pp. 23–26, 1995.

[3] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 9, pp. 2025–2041, 2002.

[4] A. Plaza, D. Valencia, J. Plaza and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral imagery," *J. Parallel Distrib. Comput.*, vol. 66, no. 3, pp. 345-358, 2006.

[5] 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.


[3] Available online from NASA's Airborne Visible Infra-Red Imaging Spectrometer's website at: http://aviris.jpl.nasa.gov/html/aviris