

Commodity Cluster-Based Parallel Implementation of an Automatic Target Generation Process for Hyperspectral Image Analysis

Sergio Bernabé and Antonio Plaza

Hyperspectral Computing Laboratory

Department of Technology of Computers and Communications

University of Extremadura, Avda. de la Universidad s/n, E-10071 Cáceres, Spain

Phone: +34 927 257000 (Ext. 51662) - Fax: +34 927 257203

E-mail: {sergiobernabe,aplaza}@unex.es

Abstract—The development of parallel implementations is an important task for hyperspectral data exploitation. In most cases, real-time or nearly real-time processing of hyperspectral images is required for swift decisions which depend upon high computing performance of algorithm analysis. A popular algorithm in hyperspectral image interpretation is the automatic target generation process (ATGP). In this paper, we develop a new parallel version of this algorithm, which is routinely applied in many application domains, including defence and intelligence, precision agriculture, geology, or forestry. We improve considerably the computational cost of this algorithm, and also improve its detection accuracy by incorporating a new method for calculating the orthogonal projection process in which the algorithm is based using the Gram-Schmidt method. Our proposed strategy reduces the computational cost over the a previous implementation of the same algorithm which uses the pseudoinverse operation to compute the orthogonal projection. Our parallel algorithm is implemented on a multi-core cluster system made up of sixteen nodes, with two CPUs of four cores per node, and quantitatively evaluated using hyperspectral data collected by NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) over the World Trade Center (WTC) in New York and over the Cuprite mining district, Nevada, United States.

Keywords—Hyperspectral imaging, automatic target generation process (ATGP), cluster computing, Gram-Schmidt orthogonalization.

I. INTRODUCTION

Hyperspectral imaging [1] 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. Hyperspectral imaging instruments such as the NASA Jet Propulsion Laboratory's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) [2] are now able to record the visible and near-infrared spectrum (wavelength region from 0.4 to 2.5 micrometers) of the reflected light of an area 2 to 12 kilometers wide and several kilometers long using 224 spectral bands [3]. The resulting "image cube" (see Fig. 1) is a stack of images in which each pixel (vector) has an associated spectral signature or fingerprint that uniquely characterizes the underlying objects. The resulting data volume typically comprises several GBs per flight [4].

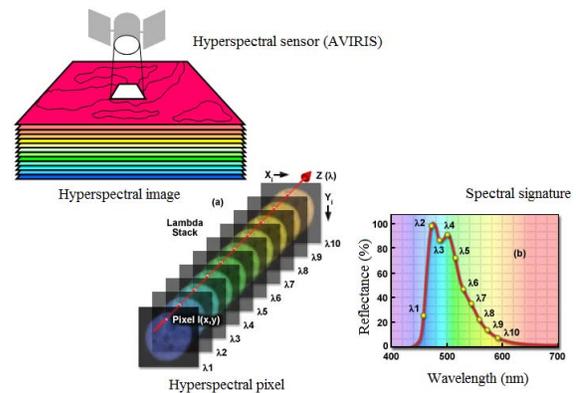


Figure 1. Concept of hyperspectral imaging.

The special properties of hyperspectral data have significantly expanded the domain of many analysis techniques. In particular, algorithms for detecting (moving or static) targets, or targets that could expand their size (such as propagating fires) often require timely responses for swift decisions that depend upon high computing performance of algorithm analysis [5]. These algorithms are considered very important tasks for hyperspectral data exploitation in defense and security applications [6].

Despite the growing interest in parallel hyperspectral imaging research [7], [8] only a few parallel implementations of automatic target detection algorithms for hyperspectral data exist in the open literature [4]. However, with the recent explosion in the amount and dimensionality of hyperspectral imagery, parallel processing is expected to become a requirement in most remote sensing missions.

In this paper, we develop a new parallel version of a well-known technique for hyperspectral data exploitation: the automatic target generation process (ATGP) [9]. The proposed technique has been implemented in a cluster-based system and makes use of the Gram-Schmidt method for orthogonalization, providing real-time results in the analysis of several hyperspectral scenes. Specifically, the parallel algorithm has been quantitatively evaluated using

hyperspectral data collected by the AVIRIS system [2] over the World Trade Center (WTC) in New York, five days after the terrorist attacks that collapsed the two main towers in the WTC complex and other case over the Cuprite mining district, Nevada, United States. The precision of the algorithms is evaluated by quantitatively substantiating their capacity to automatically detect the thermal hot spot of fires in the WTC area and the detection of mineral components in the Cuprite area. Combined, these parts offer a thoughtful perspective on the potential and emerging challenges in the design of parallel target detection algorithms using high performance computing architectures.

II. METHOD

In this section we briefly describe two versions of the ATGP target detection algorithm that will be efficiently implemented in parallel (using high-performance computing architectures) in this work. These versions are the ATGP using an orthogonal subspace projector (ATGP-OSP) and the ATGP using the Gram-Schmidt method (ATGP-GS) for calculating the orthogonal projection without using the pseudoinverse operation.

A. ATGP using an orthogonal subspace projector

ATGP was initially developed to find spectral signatures using orthogonal projections [9]. Let \mathbf{x}_0 be an initial target signature (i.e., the pixel vector with maximum length in the original n -dimensional hyperspectral image $\mathbf{F} \in \mathbf{R}^n$). This algorithm uses an orthogonal projection operator which is given by the following expression:

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T, \quad (1)$$

where \mathbf{U} is a matrix of spectral signatures, \mathbf{U}^T is the transpose of this matrix, and \mathbf{I} is the identity matrix. This orthogonal projection operator is applied to all image pixels, with $\mathbf{U} = [\mathbf{x}_0]$. It then finds a target signature, denoted by \mathbf{x}_1 , with the maximum projection in $\langle \mathbf{x}_0 \rangle^{\perp}$, which is the orthogonal complement space linearly spanned by \mathbf{x}_0 . A second target signature \mathbf{x}_2 can then be found by applying another orthogonal subspace projector $P_{\mathbf{U}}^{\perp}$ with $\mathbf{U} = [\mathbf{x}_0, \mathbf{x}_1]$ to the original image, where the target signature that has the maximum orthogonal projection in $\langle \mathbf{x}_0, \mathbf{x}_1 \rangle^{\perp}$ is selected as \mathbf{x}_2 . The above procedure is repeated until a set of target pixels $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t\}$ is extracted, where t is an input parameter to the algorithm.

B. ATGP using the Gram-Schmidt method

The purpose of using the Gram-Schmidt method in combination with the ATGP algorithm is to orthogonalize a set of vectors in an inner product space, usually the space \mathbf{R}^n in which the original hyperspectral image \mathbf{F} is defined. The Gram-Schmidt process is a finite set of linearly independent vectors $\mathbf{S} = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ for $k \leq n$, and generates an orthogonal set of vectors $\mathbf{S}' = \{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ which extends

the same k -dimensional subspace of \mathbf{R}^n as \mathbf{S} . Under these assumptions, we can now define the projection operator using the following expression:

$$proj_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}, \quad (2)$$

where $\langle \mathbf{v}, \mathbf{u} \rangle$ denotes the inner product of vectors \mathbf{v} and \mathbf{u} . The Gram-Schmidt process now continues as follows:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{v}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\ \mathbf{u}_2 &= \mathbf{v}_2 - proj_{\mathbf{u}_1}(\mathbf{v}_2), & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\ \mathbf{u}_3 &= \mathbf{v}_3 - proj_{\mathbf{u}_1}(\mathbf{v}_3) - proj_{\mathbf{u}_2}(\mathbf{v}_3), & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\ \mathbf{u}_4 &= \mathbf{v}_4 - proj_{\mathbf{u}_1}(\mathbf{v}_4) - proj_{\mathbf{u}_2}(\mathbf{v}_4) - proj_{\mathbf{u}_3}(\mathbf{v}_4), & \mathbf{e}_4 &= \frac{\mathbf{u}_4}{\|\mathbf{u}_4\|} \\ &\vdots & &\vdots \\ \mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} proj_{\mathbf{u}_j}(\mathbf{v}_k), & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|} \end{aligned}$$

where the sequence $\mathbf{u}_1, \dots, \mathbf{u}_k$ is the system of orthogonal vectors required, and the normalized vectors $\mathbf{e}_1, \dots, \mathbf{e}_k$ form an orthonormal set. The calculation of the sequence $\mathbf{u}_1, \dots, \mathbf{u}_k$ is called Gram-Schmidt orthogonalization, while the calculation of the sequence $\mathbf{e}_1, \dots, \mathbf{e}_k$ is referred to as Gram-Schmidt orthonormalization (i.e. the vectors are normalized). To verify that the above equations produce an orthogonal sequence, we calculate first $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle$ substituting the above equation for \mathbf{u}_2 , we would obtain a result equal to 0. Same for calculating $\langle \mathbf{u}_1, \mathbf{u}_3 \rangle$ again replacing the equation by \mathbf{u}_3 , we obtain a result equal to 0. The general test is obtained by mathematical induction. In geometrical terms (see Fig. 2), this method proceeds as follows: to calculate \mathbf{u}_i , \mathbf{v}_i is projected orthogonally to the subspace \mathbf{U} generated by $\mathbf{u}_1, \dots, \mathbf{u}_{i-1}$, which is the same as the subspace spanned by $\mathbf{v}_1, \dots, \mathbf{v}_{i-1}$. The vector \mathbf{u}_i is defined as the difference between \mathbf{v}_i and the projection of this, ensuring that it is orthogonal to all vectors in the subspace \mathbf{U} .

III. PARALLEL IMPLEMENTATION FOR CLUSTERS OF COMPUTERS

Clusters of computers are made up of different processing units interconnected via a communication network [10]. In previous work, it has been reported that data-parallel approaches, in which the hyperspectral data is partitioned among different processing units, are particularly effective for parallel processing in this type of high performance computing systems [11], [12], [13], [14]. In this framework, it is very important to define the strategy for partitioning the hyperspectral data. In our implementations, a data-driven partitioning strategy has been adopted as a baseline for

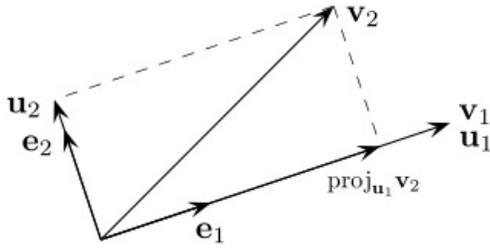


Figure 2. Geometric representation of the Gram-Schmidt process.

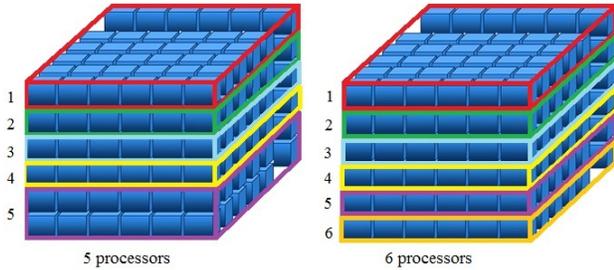


Figure 3. Spatial-domain decomposition of a hyperspectral data set into five (left) and six (right) partitions.

algorithm parallelization. Specifically, two approaches for data partitioning have been tested [5]:

- *Spectral-domain partitioning*. This approach subdivides the multi-channel remotely sensed image into small cells or sub-volumes made up of contiguous spectral wavelengths for parallel processing.
- *Spatial-domain partitioning*. This approach breaks the multi-channel image into slices made up of one or several contiguous spectral bands for parallel processing. In this case, the same pixel vector is always entirely assigned to a single processor, and slabs of spatially adjacent pixel vectors are distributed among the processing nodes of the parallel system. Fig. 3 shows two examples of spatial-domain partitioning over five processors and over six processors, respectively.

Previous experimentation with the above-mentioned strategies indicated that spatial-domain partitioning can significantly reduce inter-processor communication, resulting from the fact that a single pixel vector is never partitioned and communications are not needed at the pixel level [5]. In the following, we assume that spatial-domain decomposition is always used when partitioning the hyperspectral data cube. The inputs to the parallel algorithm are a hyperspectral image cube \mathbf{F} with n dimensions, and a maximum number of targets to be detected, t . The output in all cases is a set of t target pixel vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t\}$. The algorithm has been implemented in the C++ programming language using calls to *MPI*, the message passing interface library commonly available for parallel implementations in multi-processor

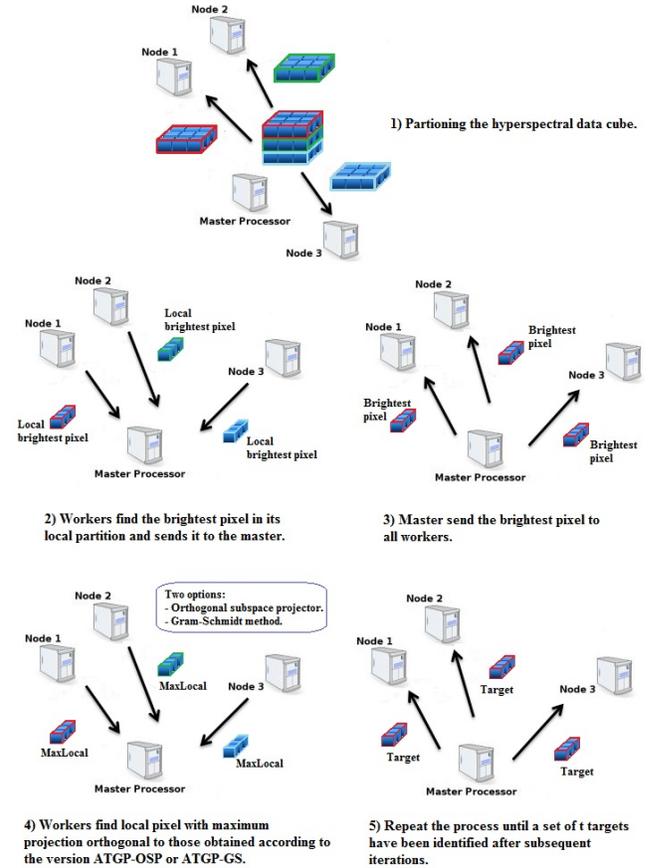


Figure 4. Graphical summary of the ATGP-OSP or ATGP-GS versions using 1 master processor and 3 slaves.

systems. The two versions: ATGP-OSP and ATGP-GS use a similar strategy summarized in Fig. 4.

IV. EXPERIMENTAL RESULTS

This section is organized as follows. First we describe the AVIRIS hyperspectral data sets used in our experiments. Then we describe the parallel computing platforms used for experimental evaluation, which is a Beowulf cluster located at CETA-Ciemat¹ supercomputer center in Extremadura. Then we discuss the target detection accuracy of the considered parallel algorithm when analyzing the two considered hyperspectral data sets. Finally we describe the parallel performance results obtained after implementing the ATGP using an orthogonal subspace projector (ATGP-OSP) and the Gram-Schmidt method (ATGP-GS) on the Beowulf cluster.

A. Data description

The two scenes used for experiments in this work were collected by the AVIRIS instrument. For the first image, the instrument was flown by NASA's Jet Propulsion Laboratory over the World Trade Center (WTC) area in New

¹<http://www.ceta-ciemat.es>



Figure 5. False color composition of an AVIRIS hyperspectral image collected by NASA's Jet Propulsion Laboratory over lower Manhattan on Sept. 16, 2001 (left). Location of thermal hot spots in the World Trade Center area, available online: <http://pubs.usgs.gov/of/2001/ofr-01-0429/hotspot.key.tgif> (right).

York City on September 16, 2001, just five days after the terrorist attacks that collapsed the two main towers and other buildings in the WTC complex and secondly over the well-known AVIRIS Cuprite data set, available online in reflectance units after atmospheric correction. The data set consists of 614×512 pixels, 224 spectral bands and a total size of (approximately) 140 MB. The spatial resolution is 1.7 meters per pixel. The leftmost part of Fig. 5 shows a false color composite of the data set selected for experiments using the 1682, 1107 and 655 nm channels, displayed as red, green and blue, respectively. Vegetated areas appear green in the leftmost part of Fig. 5, while burned areas appear dark gray. Smoke coming from the WTC area (in the red rectangle) and going down to south Manhattan appears bright blue due to high spectral reflectance in the 655 nm channel. Extensive reference information, collected by U.S. Geological Survey (USGS), is available for the WTC scene. In this work, we use a U.S. Geological Survey thermal map which shows the target locations of the thermal hot spots at the WTC area, displayed as bright red, orange and yellow spots at the rightmost part of Fig. 5. The map is centered at the region where the towers collapsed, and the temperatures of the targets range from 700F to 1300F. The thermal map displayed in the rightmost part of Fig. 5 will be used in this work as groundtruth to validate the target detection accuracy of the proposed parallel algorithms and their respective serial versions.

A second hyperspectral image scene has been considered for experiments. It is the well-known AVIRIS Cuprite scene (see Fig. 6(a)), collected in the summer of 1997 and available online in reflectance units after atmospheric correction. The portion used in experiments corresponds to a 350×350 -pixel subset of the sector labeled as f970619t01p02_r02_sc03.a.rfi in the online data, which comprises 188 spectral bands in the range from 400 to 2500 nm and a total size of around 50 MB. Water absorption and low SNR bands were removed prior to

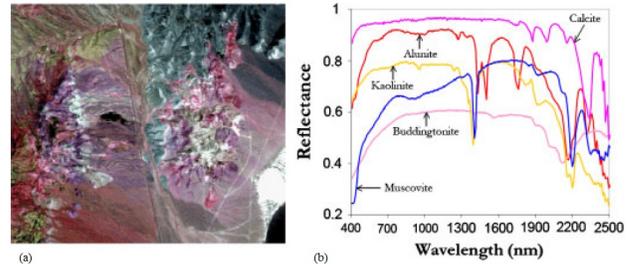


Figure 6. (a) False color composition of the AVIRIS hyperspectral over the Cuprite mining district in Nevada and (b) U.S. Geological Survey mineral spectral signatures used for validation purposes.

the analysis. The site is well understood mineralogically, and has several exposed minerals of interest, including alunite, buddingtonite, calcite, kaolinite, and muscovite. Reference ground signatures of the above minerals (see Fig. 6(b)), available in the form of a USGS library will be used to assess ATGP algorithm in this work.

B. Parallel computing platform

The parallel computing architecture used in experiments is composed of 2.26 GHz Intel Xeon nodes, each with 12 GB of SDRAM DDR3 memory and a scratch area of 1TB of memory shared among the different processors. The total peak performance of the system is 16 Tflops. This cluster is composed of a total of 128 cores. The parallel algorithms tested in this work were run from one of such nodes. The operating system used at the time of experiments was Scientific Linux CERN SLC release 4.6 (Beryllium), and MPICH2 was the message-passing library used.

C. Analysis of target detection accuracy

It is first important to emphasize that our parallel versions of ATGP provides exactly the same results as the serial versions of the same algorithms, implemented using the Intel C/C++ and gcc (gnu compiler default) compilers and optimized via compilation flags to exploit data locality and avoid redundant computations. These results were also exactly the same as those achieved by the serial implementation and, hence, the only difference between the considered versions (serial and parallel) is the time they need to complete their calculations, which varies depending on the computer architecture in which they are run. In the following we describe the experimental results obtained for the two considered AVIRIS scenes:

1) *Results with the AVIRIS WTC scene:* Table I shows the spectral angle distance (SAD) [3] values (in degrees) between the most similar target pixels detected by ATGP-OSP (this version uses an orthogonal subspace projector) and the pixel vectors at the known target positions, labeled from A to H in the rightmost part of Fig. 5. The same results for the version ATGP-GS (this version uses the Gram-Schmidt method) are also reported. In all cases, the number of target

Table I
SPECTRAL ANGLE VALUES (IN DEGREES) BETWEEN TARGET PIXELS AND KNOW GROUND TARGETS FOR ATGP-OSP AND ATGP-GS OVER THE WTC SCENE.

Version	A	B	C	D	E	F	G	H
ATGP-OSP	0.00°	14.43°	0.00°	27.38°	20.32°	7.13°	4.15°	31.27°
ATGP-GS	0.00°	27.16°	0.00°	15.62°	27.81°	3.98°	2.72°	24.26°

Table II
SPECTRAL ANGLE VALUES (IN DEGREES) BETWEEN TARGET PIXELS AND KNOW GROUND TARGETS FOR ATGP-OSP AND ATGP-GS OVER THE CUPRITE SCENE.

Version	Alunite	Budding.	Calcite	Kaolinite	Muscovite	Average
ATGP-OSP	4.81°	4.16°	9.52°	10.76°	5.29°	6.91°
ATGP-GS	5.48°	4.08°	5.87°	11.14°	5.68°	6.45°

pixels to be detected was set to $t = 30$ after calculating the virtual dimensionality (VD) of the data [15]. As shown by Table I, the ATGP-OSP and ATGP-GS extracted targets which were similar, spectrally, to the known ground-truth targets (this versions were able to perfectly detect the targets labeled as A and C, and had more difficulties in detecting very small targets).

2) *Results for the AVIRIS Cuprite scene:* Table II shows the SAD values (in degrees) between the most similar target pixels detected by ATGP-OSP (this version uses an orthogonal subspace projector) and the pixel vectors at the known target minerals, labeled as alunite, buddingtonite, calcite, kaolinite, and muscovite in the rightmost part of Fig. 6. The same results are reported for the ATGP-GS. In all cases, the number of target pixels to be detected was set to $t = 19$ after calculating the VD. As shown by Table II, again the ATGP-OSP and ATGP-GS extracted targets were similar, spectrally, to the known ground-truth targets, although ATGP-GS version offers a better result in terms of average.

D. Parallel performance

In this subsection we evaluate the parallel performance of ATGP-OSP and ATGP-GS in the considered Beowulf cluster. Table III and Table IV respectively show the processing times in seconds the multi-processor version of ATGP-OSP and ATGP-GS using different numbers of processors on the Beowulf cluster. As shown by Table III, when 16 cores were used the ATGP-OSP was able to finalize in about 11 seconds and the version ATGP-GS was able to finalize in about 1.2 seconds. This result is strictly in real-time as the AVIRIS instrument takes about 5 seconds to collect the AVIRIS WTC scene. Thus, the parallel versions clearly outperform the respective sequential versions which take over two and half minutes and 14.5 seconds, respectively. On the other hand, Table IV reveals that ATGP-OSP was able to finalize in about 2 seconds (the time needed for AVIRIS to collect the Cuprite scene) and the version ATGP-GS was able to finalize in about 0.3 seconds with 16 cores. These results are also in real-time.

Table III
PROCESSING TIMES IN SECONDS MEASURED FOR ATGP-OSP AND ATGP-GS USING DIFFERENT NUMBERS OF CORES ON THE CONSIDERED CLUSTER AFTER PROCESSING THE AVIRIS WTC SCENE.

Version	1 CORE	2 CORES	4 CORES	8 CORES	16 CORES
ATGP-OSP	153.65	77.47	38.85	20.43	10.93
ATGP-GS	14.50	7.32	3.68	1.85	1.24

Table IV
PROCESSING TIMES IN SECONDS MEASURED FOR ATGP-OSP AND ATGP-GS USING DIFFERENT NUMBERS OF CORES ON THE CONSIDERED CLUSTER AFTER PROCESSING THE AVIRIS CUPRITE SCENE.

Version	1 CORE	2 CORES	4 CORES	8 CORES	16 CORES
ATGP-OSP	27.23	13.83	6.99	3.71	2.01
ATGP-GS	3.141	1.58	0.79	0.40	0.27

Table V
SPEEDUP ACHIEVED BY THE ATGP-OSP AND ATGP-GS ALGORITHMS USING DIFFERENT NUMBERS OF CORES ON THE CONSIDERED CLUSTER AFTER PROCESSING THE AVIRIS WTC SCENE.

Version	2 CORES	4 CORES	8 CORES	16 CORES
ATGP-OSP	1.983	3.955	7.521	14.057
ATGP-GS	1.982	3.939	7.821	11.685

Table VI
SPEEDUP ACHIEVED BY THE ATGP-OSP AND ATGP-GS ALGORITHMS USING DIFFERENT NUMBERS OF CORES ON THE CONSIDERED CLUSTER AFTER PROCESSING THE AVIRIS CUPRITE SCENE.

Version	2 CORES	4 CORES	8 CORES	16 CORES
ATGP-OSP	1.969	3.896	7.335	13.532
ATGP-GS	1.982	3.954	7.925	11.606

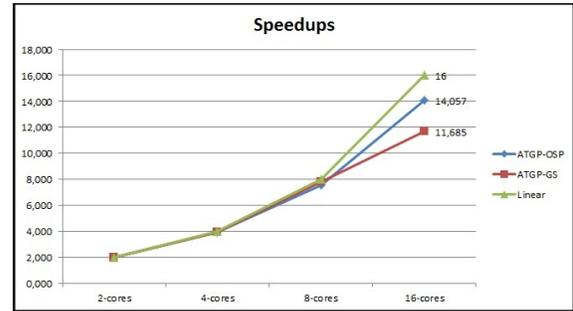


Figure 7. Graphical representation of the speedups achieved by the ATGP-OSP and ATGP-GS algorithms using different numbers of cores on the considered cluster after processing the AVIRIS WTC scene.

Finally, Tables V and VI respectively report the speedups (number of times that the parallel version was faster than the sequential one as the number of processors was increased) achieved by multi-processor runs of the ATGP-OSP and ATGP-GS algorithms for the AVIRIS WTC and Cuprite scenes. For illustrative purposes, the speedups are also graphically illustrated in Fig. 7 and Fig. 8, which indicate that ATGP-OSP and ATGP-GS resulted in speedups close to linear in both cases.

V. CONCLUSIONS AND FUTURE RESEARCH

This paper described two parallel versions of an automatic target generation process (ATGP) for hyperspectral image

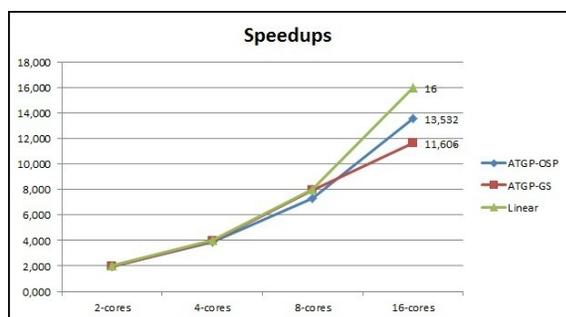


Figure 8. Graphical representation of the speedups achieved by the ATGP-OSP and ATGP-GS algorithms using different numbers of cores on the considered cluster after processing the AVIRIS Cuprite scene.

analysis. We have investigated the impact of including Gram-Schmidt method (ATGP-GS) for calculating orthogonal projections instead of using an orthogonal subspace projector (ATGP-OSP) that includes a very expensive operation, i.e. the calculation of the inverse, in the traditional implementation of this algorithm. The parallel versions have been implemented in a Beowulf cluster. Experimental results, oriented towards analyzing the target detection accuracy and parallel performance of the proposed parallel versions, have been presented and thoroughly discussed in the context of two real applications: the analysis of hyperspectral data collected by NASA's AVIRIS instrument over the World Trade Center (WTC) in New York, five days after the terrorist attacks that collapsed the two main towers in the WTC complex and over the Cuprite mining district, Nevada, United States. The results obtained reveal that ATGP-GS provides better results than the version ATGP-OSP, achieving a response in real-time. Currently we are experimenting with FPGA platforms in order to be able to adapt the proposed algorithms to hardware devices that can be mounted onboard hyperspectral imaging instruments after certification by international agencies.

ACKNOWLEDGMENT

This work has been supported by the European Community's Marie Curie Research Training Networks Programme under reference MRTN-CT-2006-035927 (HYPER-I-NET). Funding from the Spanish Ministry of Science and Innovation (HYPERCOMP/EODIX project, reference AYA2008-05965-C04-02) is also gratefully acknowledged.

REFERENCES

- [1] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for Earth remote sensing," *Science*, vol. 228, pp. 1147–1153, 1985.
- [2] 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, vol. 65, no. 3, pp. 227–248, 1998.

- [3] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Kluwer Academic/Plenum Publishers: New York, 2003.
- [4] A. Plaza and C.-I. Chang, *High Performance Computing in Remote Sensing*. Taylor & Francis: Boca Raton, FL, 2007.
- [5] 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, no. 3, pp. 345–358, 2006.
- [6] A. Plaza and C.-I. Chang, "Clusters versus FPGA for parallel processing of hyperspectral imagery," *International Journal of High Performance Computing Applications*, vol. 22, no. 4, pp. 366–385, 2008.
- [7] A. Plaza, J. Plaza, and D. Valencia, "Impact of platform heterogeneity on the design of parallel algorithms for morphological processing of high-dimensional image data," *Journal of Supercomputing*, vol. 40, pp. 81–107, 2007.
- [8] 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*, vol. 43, pp. 441–445, 2007.
- [9] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232–1249, 2003.
- [10] R. Brightwell, L. Fisk, D. Greenberg, T. Hudson, M. Levenhagen, A. Maccabe, and R. Riesen, "Massively parallel computing using commodity components," *Parallel Computing*, vol. 26, p. 243266, 2000.
- [11] A. Plaza, J. Plaza, and D. Valencia, "Impact of platform heterogeneity on the design of parallel algorithms for morphological processing of high-dimensional image data," *Journal of Supercomputing*, vol. 40, no. 1, pp. 81–107, 2007.
- [12] A. Plaza, D. Valencia, and J. Plaza, "An experimental comparison of parallel algorithms for hyperspectral analysis using homogeneous and heterogeneous networks of workstations," *Parallel Computing*, vol. 34, no. 2, pp. 92–114, 2008.
- [13] A. Paz and A. Plaza, "Clusters versus GPUs for parallel automatic target detection in remotely sensed hyperspectral images," *EURASIP Journal on Advances in Signal Processing*, vol. 915639, pp. 1–18, 2010.
- [14] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel hyperspectral image and signal processing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 119–126, 2011.
- [15] C.-I. Chang and Q. Du, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 608–619, 2004.