Proceedings of the 10th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2010 27–30 June 2010.

A New Parallel Implementation of the RX Algorithm for Anomaly Detection in Hyperspectral Images

A. Paz[2], J. M. Molero[1], E. M. Garzón[1], J. A. Martínez[1] and A. Plaza[2]

[1] *Department of Computer Architecture and Electronics, University of Almería*

[2] *Department of Technology of Computers and Communications, University of Extremadura*

emails: apazgal@unex.es, jmolero@ace.ual.es, gmartin@ual.es, jamartine@ual.es, aplaza@unex.es

Abstract

Remotely sensed hyperspectral sensors provide image data containing rich information in both the spatial and the spectral domain, and this information can be used to address detection tasks in many applications. In many surveillance applications, the size of the objects (targets) searched for constitutes a very small fraction of the total search area and the spectral signatures associated to the targets are generally different from those of the background, hence the targets can be seen as anomalies. One of the most widely used and successful algorithms for anomaly detection in hyperspectral images is the one proposed by Reed and Xiaoli, commonly known as RX algorithm. Despite its wide acceptance and high computational complexity when applied to real hyperspectral scenes, few approaches have been developed for parallel implementation of this algorithm due to the complex calculation of the inverse of the sample covariance matrix in parallel. In this paper, we evaluate the suitability o using a local approach for the calculation of the inverse of the sample covariance matrix of a high-dimensional hyperspectral scene in parallel. The considered approach is quantitatively evaluated using hyperspectral data collected by the NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) system 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. The precision of the algorithms is evaluated by quantitatively substantiating their capacity to automatically detect the thermal hot spot of fires (anomalies) in the WTC area.

Key words: Hyperspectral imaging, anomaly detection, Reed-Xiaoli algorithm (RX).

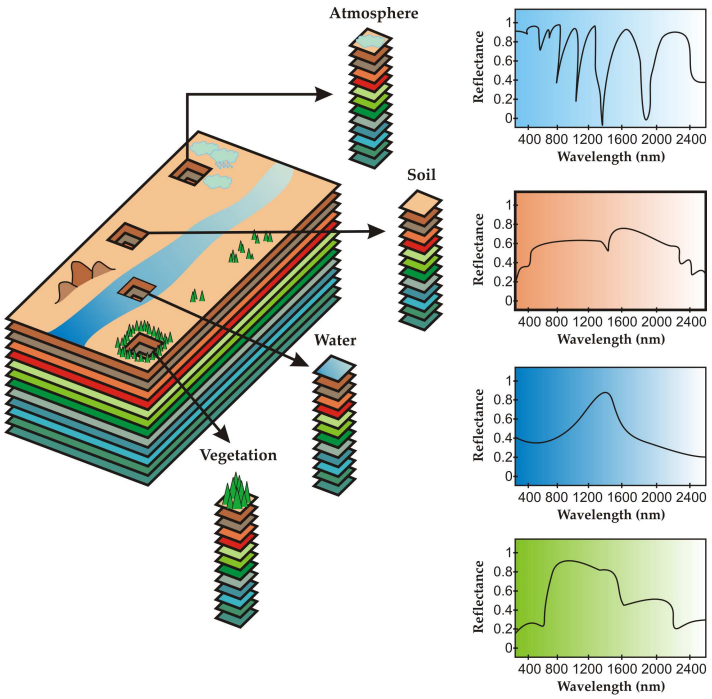

Figure 1: Concept of hyperspectral imaging.

1 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 [2]. Hyperspectral imaging instruments such as the NASA Jet Propulsion Laboratory's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) [3] 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. 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 [4]. The resulting data volume typically comprises several GBs per flight [5].

Anomaly detection is an important task for hyperspectral data exploitation. An anomaly detector enables one to detect spectral signatures which are spectrally distinct from their surroundings with no *a priori* knowledge. In general, such anomalous signatures are relatively small compared to the image background, and only occur in the image with low probabilities. A well-known approach for anomaly detection was developed by Reed and Yu, and is referred to as the RX algorithm, which has shown success in anomaly detection for multispectral and hyperspectral images [4]. The RX uses the pixel currently being processed as the matched signal. Since the RX uses the sample covariance matrix to take into account the sample spectral correlation, it performs the

same task as the Mahalanobis distance, which has been widely used in hyperspectral imaging applications [6]. A variation of the algorithm consists in applying the same concept in local neighborhoods centered around each image pixel, this is known as the kernel version of the RX algorithm [7].

Despite its wide acceptance and high computational complexity when applied to real hyperspectral scenes, few approaches have been developed for parallel implementation of this algorithm due to complexity of calculating the sample covariance matrix (and its inverse) in parallel. The success of the kernel version of the algorithm in [7] led us to believe that a standard data partitioning framework for parallel implementation may provide different (or even better) results if the sample covariance matrices are calculated independently for small data portions rather than computing the sample covariance matrix for the entire hyperspectral image, which requires additional inter-processor communications that may reduce parallel performance. This aspect is crucial for the RX implementation since the consideration of a local or global strategy for the computation of the sample covariance matrix is expected to show an important impact in the scalability of the parallel solution but there is a trade-off between the increase in parallel efficiency and the quality of the final solution in terms of anomaly detection accuracy.

In this paper, we investigate the use of a local approach for the calculation of the inverse of the sample covariance matrix, in which each processing node calculates the covariance matrix of its local partition in parallel and inter-processor communications are significantly reduced. The remainder of the paper is structured as follows. Section 2 briefly describes the classic RX algorithm. Section 3 describes the parallel implementation adopted in this work. Section 4 describes the hyperspectral data set considered in experiments, which comprises a data set collected by the NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) system 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. Section 5 conducts a detailed experimental assessment of the precision and scalability of the proposed parallel implementations using the aforementioned scene as a relevant case study. Finally, section 6 concludes with some remarks and hints at plausible future research.

2 RX algorithm

The RX algorithm has been widely used in signal and image processing [8]. The filter implemented by this algorithm is referred to as RX filter and defined by the following expression:

$$\delta^{\mathrm{RX}}(\mathbf{x}) = (\mathbf{x} - \mu)^T \mathbf{K}^{-1} (\mathbf{x} - \mu), \tag{1}$$

where $\mathbf{x} = \left[x^{(0)}, x^{(1)}, \cdots, x^{(n)}\right]$ is a sample, n-dimensional hyperspectral pixel (vector), μ is the sample mean of the hyperspectral image and $\mathbf{K}$ is the sample data covariance matrix. As we can see, the form of δ^{RX} is actually the well-known Mahalanobis distance [6]. It is important to note that the images generated by the RX algorithm are generally

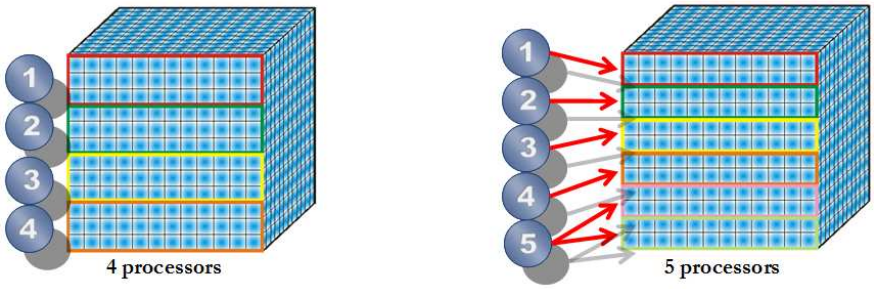

Figure 2: Spatial-domain decomposition of a hyperspectral data set.

gray scale images. In this case, the anomalies can be categorized in terms of the value returned by RX, so that the pixel with higher value of $\delta^{\mathrm{RX}}(\mathbf{x})$ can be considered the first anomaly, and so on.

3 Parallel implementation

3.1 Data partitioning strategies

In the considered parallel algorithms, a data-driven partitioning strategy has been adopted as a baseline for algorithm parallelization. Specifically, two approaches for data partitioning have been tested [9]:

- *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 (CPUs) of the parallel system. Fig. 2 shows two examples of spatial-domain partitioning over 4 processors and over 5 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 [9]. In the following, we assume that spatial-domain decomposition is always used when partitioning the hyperspectral data cube.

3.2 Parallel algorithm

Our parallel version of the RX algorithm for anomaly detection adopts the spatial-domain decomposition strategy depicted in Fig. 2 for dividing the hyperspectral data

cube in master-slave fashion. The approach considered in this work represents a variation of the one presented in [10, 11], in which a global approach is adopted for the covariance matrix calculation. In this work, we use a local approach for the calculation of the covariance matrix instead, i.e. each node calculates the covariance matrix of its local partition. The parallel algorithm is given by the following steps:

1. The master processor divides the original image cube into P spatial-domain partitions and distributes them among the workers.

2. The master calculates the n-dimensional mean vector $\mathbf{m}$ concurrently, where each component is the average of the pixel values of each spectral band of the unique set. This vector is formed once all the processors finish their parts.

3. Each worker calculates the covariance matrix of its local partition, and applies (locally) the RX filter given by the Mahalanobis distance to all the pixel vectors in the local partition as follows: $\delta^{(\mathrm{RX})}(\mathbf{x}) = (\mathbf{x} - \mathbf{m})^T \mathbf{K}^{-1} (\mathbf{x} - \mathbf{m})$, and returns the local result to the master.

4. The master now selects the t pixel vectors with higher associated value of $\delta^{(\mathrm{RX})}$, and uses them to form a final set of targets $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t\}$.

4 Hyperspectral data set

The image scene used for experiments in this work was collected by the AVIRIS instrument, which was flown by NASA's Jet Propulsion Laboratory over the World Trade Center (WTC) area in New 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. The full data set selected for experiments 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. 3 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. 3, 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[1]. In this work, we use a U.S. Geological Survey thermal map[2] which shows the locations of the thermal hot spots (which can be seen as anomalies) at the WTC area, displayed as bright red, orange and yellow spots at the rightmost part of Fig. 3. The map is centered at the region where the towers collapsed, and the temperatures of the targets range from 700F to 1300F. Further information available from USGS about the thermal hot spots (including location and temperature)

[1] http://speclab.cr.usgs.gov/wtc

[2] http://pubs.usgs.gov/of/2001/ofr-01-0429/hotspot.key.tgif.gif

Figure 3: 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 fires observed in World Trade Center area, available online: http://pubs.usgs.gov/of/2001/ofr-01-0429/hotspot.key.tgif.gif (right).

is reported on Table 1. The thermal map displayed in the rightmost part of Fig. 3 will be used in this work as ground-truth to validate the target detection accuracy of the proposed parallel algorithms and their respective serial versions.

5 Evaluation

The parallel computing platform used in this experiments is the Sun Fire x4600, it is composed of 8 quad 2.3 GHz AMD Opteron 8356 (32 cores), with 64 Gb of main memory. The operating system used at the time of experiments was Debian, and Open MPI was used as parallel interface programming. Although the selected parallel

Table 1: Properties of the thermal hot spots reported in the rightmost part of Fig. 3.

Hot spot	**Latitude (North)**	**Longitude (West)**	**Temperature (Kelvin)**
'A'	$40^o42'47.18"$	$74^o00'41.43"$	1000
'B'	$40^o42'47.14"$	$74^o00'43.53"$	830
'C'	$40^o42'42.89"$	$74^o00'48.88"$	900
'D'	$40^o42'41.99"$	$74^o00'46.94"$	790
'E'	$40^o42'40.58"$	$74^o00'50.15"$	710
'F'	$40^o42'38.74"$	$74^o00'46.70"$	700
'G'	$40^o42'39.94"$	$74^o00'45.37"$	1020
'H'	$40^o42'38.60"$	$74^o00'43.51"$	820

platform is based on a shared memory architecture, according to our experience MPI has shown how is able to exploit this architecture and adapts to the characteristic of our problem [13].

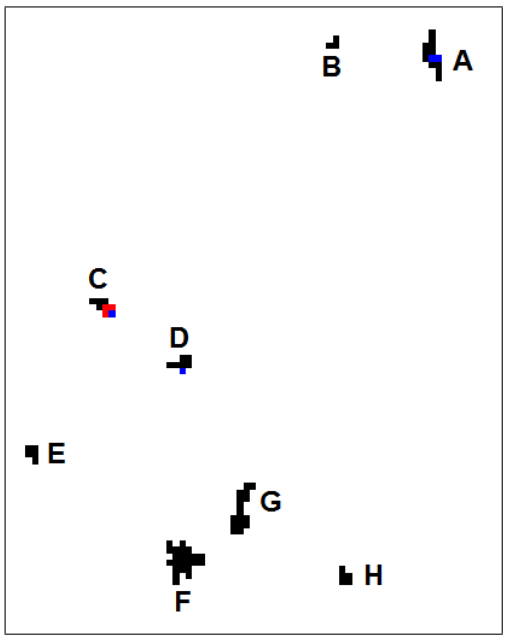

Figure 4: Detection results for the different implementations of RX, where black pixels represent the ground-truth, red pixels represent the targets detected by the serial version, and blue pixels represent the targets detected by the parallel version.

The evaluation of RX capability to automatically detect the anomalies is based on the results showed in Fig. 4. For illustrative purposes, Fig. 4 shows the detection results obtained by the serial and the parallel implementation of RX using different number of processors ($P = 16$ and $P = 32$). In all cases, the number of target pixels to be detected was set to $t = 30$ after calculating the virtual dimensionality of the data [4]. The pixels labeled in black color in Fig. 4 represent the original targets. The pixels labeled in red color in Fig. 4 represent the pixels detected by the serial version. Finally, the pixels labeled in blue color in Fig. 4 represent the pixels detected by the parallel version (the results obtained using $P = 16$ and $P = 32$ processors were overlapped). As shown by Fig. 4, the same targets were detected by the parallel version regardless of the number of partitions. In this case, a local approach is used for calculating the covariance matrix, while the serial version uses a global approach. The global strategy introduces additional inter-processor communications which negatively affect parallel performance as indicated in [10, 11], while the local strategy exhibits results which are almost identical to those obtained by the global strategy. In all cases, only three out of eight targets (i.e. those labeled as 'A', 'C' and 'D') were detected. However, increasing the number of targets to be detected t increases the detection results. In this work, however, we have decided to adopt the value $t = 30$ based on the calculation of the

virtual dimensionality of the data, which provides an objective criterion for setting the number of targets.

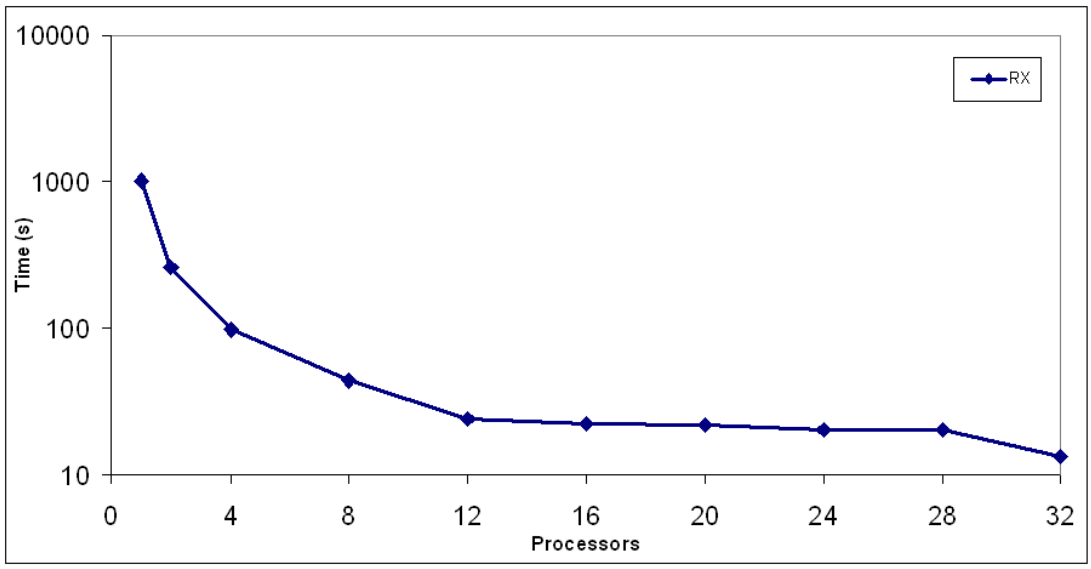

Figure 5: Run-Time of parallel RX.

Fig. 5 shows the performance of parallel RX for different number of processors P. It shows a super-linear speed-up because $T(1)/T(P) > P$ for all values of P. Experiences with super-linear speed-up in parallel computing have been described in the literature [12, 14]. The super-linear speed-up is achieved when the computational power increases as the load of every processor decreases. Frequently, this behavior is related to applications which need large memory requirements, since the memory management is improved as P increases. The memory requirements to store the hyperspectral images are large. Consequently, the parallel RX, based on the partition of hyperspectral image, shows superlinear speed-up due to the improvements of the memory management when P increases.

Hence, the speed-up is not an appropriate parameter to evaluate the scalability of parallel RX, since the sequential RX is penalized by a hard memory management. With this aim, the parameter called Incremental Speed-up ($IncSpUp$) has been proposed as an alternative to the speed-up [12]. It provides information about how much the computing time diminishes when the number of PEs increases. $IncSpUp$ is defined as follows:

$$IncSpUp(2^k) = \frac{T(P = 2^{k-1})}{T(P = 2^k)}, \tag{2}$$

where T(P) is the run time of the execution with P PEs. Values of the Incremental Speed-up equal to 2 are equivalent to lineal speed-ups.

Figure 6 shows $IncSpUp$ obtained by the parallel RX algorithm using different number of processors on the considered parallel system. As shown by Figure 6, the

proposed parallel implementation provides good scalability results when compared to the implementation in [10, 11]. These results are a consequence of a key of parallel RX, that is, there are not communications among the processors which penalize the performance when P increases. However, the proposed parallel implementation of the RX algorithm can still be enhanced by further exploring different strategies for the parallel calculation of the covariance matrix.

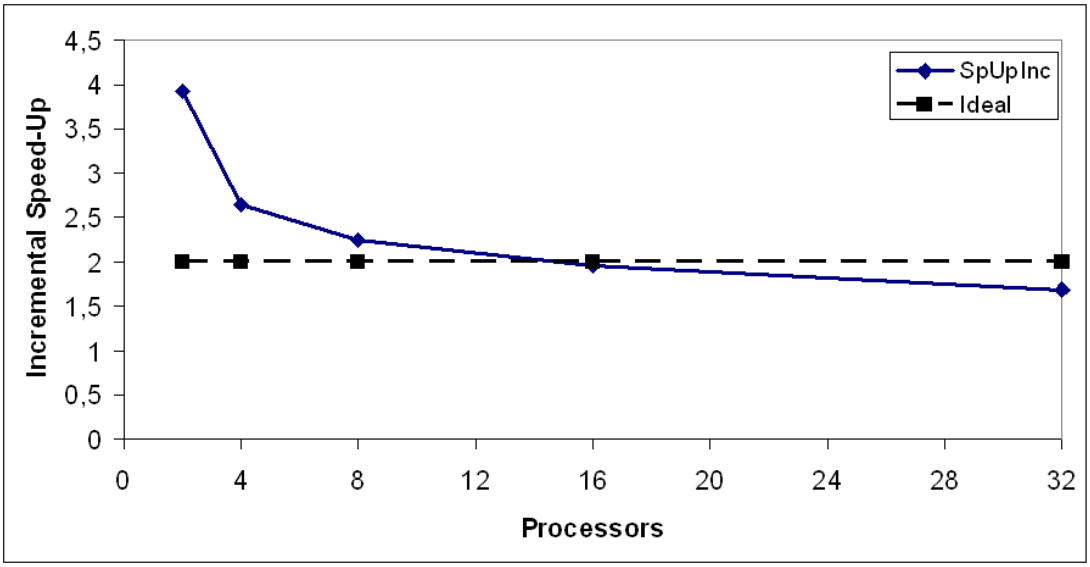

Figure 6: Incremental Speedup for the RX algorithm.

6 Conclusions and future research lines

In this paper, we have analyzed the accuracy and scalability of a new parallel implementation of the RX algorithm which uses a local approach for the calculation of the covariance matrix in parallel. The proposed approach has some competitive advantages (in terms of scalability) with regards to the commonly used (global) strategies adopted for calculating the covariance matrix in parallel when implementing this algorithm. The parallel version has been validated in the context of a real hyperspectral imaging application, focused on detecting the thermal hot spots (anomalies) of the fires in the World Trade Center area in New York City, just few days after the terrorist attacks of September 11th, 2001. Our experimental results indicate that the proposed (local) strategy provides similar accuracy results to those reported for the (global) one adopted in previous work, and presents potential for improving the scalability of the algorithm due to the fact that the proposed strategy reduces significantly the amount of inter-processor communications. Although the results reported in this work are encouraging, further experiments should be conducted in order to increase the scalability of the proposed parallel algorithms to a higher number of processors by resolving memory issues

and optimizing the parallel design of such algorithms. Experiments with additional scenes under different target/anomaly detection scenarios are also highly desirable.

Acknowledgements

This work has been supported by the European Communitys Marie Curie Research Training Networks Programme under reference MRTN-CT-2006-035927 (HYPER-I-NET) and by Spanish Ministry of Science and Innovation TIN2008-01117. Funding from the Spanish Ministry of Science and Innovation (HYPERCOMP/EODIX project, reference AYA2008-05965-C04-02) is 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] 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*, vol. 113, pp. 110–122, 2009.

[3] 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.

[4] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification.* Norwell, MA: Kluwer, 2003.

[5] A. Plaza and C.-I. Chang, *High performance computing in remote sensing.* Boca Raton: CRC Press, 2007.

[6] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction.* Springer, 2006.

[7] H. Kwon and N. M. Nasrabadi, "Hyperspectral anomaly detection using kernel rx-algorithm." in *International Conference on Image Processing (ICIP)*, 2004.

[8] I. Reed and X.Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distrihution." *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 38, pp. 1760–1770, 1990.

[9] 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.

[10] A. Paz, A. Plaza, and S. Blazquez, "Parallel implementation of target and anomaly detection algorithms for hyperspectral imagery," *Proc. IEEE Geosci. Remote Sens. Symp.*, vol. 2, pp. 589–592, 2008.

[11] 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," *Proc. SPIE*, vol. 7455, pp. 1–12, 2009.

[12] Garzón, E.M. and García, I. "A parallel implementation of the eigenproblem for large, symmetric and sparse matrices," *Recent advances in PVM and MPI, LNCS 1697*, Springer-Verlag, 1999, 380–387.

[13] S.Tabik, E.M. Garzón, I. García, and J.J. Fernández. High Performance Noise Reduction for Biomedical Multidimensional Data.*Digital Signal Processing*, Vol. 17, n. 4, pp. 724–736. 2007.

[14] Gustafson, J.L. Reevaluating Amdahl's Law, *Communications of the ACM*, 1988, 532–533