FPGA Design of the N-FINDR Algorithm for Spaceborne Hyperspectral Missions

Carlos González, Daniel Mozos
Departament of Computer Architecture and Automatics
Computer Science Faculty
Complutense University of Madrid
Emails: carlosgonzalez@fdi.ucm.es, mozos@fis.ucm.es

Javier Resano
Departament of Computer Architecture
High Polytechnic Center
University of Zaragoza
Email: jresano@unizar.es

Antonio Plaza
Departament of Computer Technology and Communications
Polytechnic School of Cáceres
University of Extremadura
Email: aplaza@unex.es

***Abstract*—Hyperspectral imaging is a new technique in remote sensing which generates hundreds of images (at different wavelength channels) for the same area on the surface of the Earth. Each pixel collected by a hyperspectral remote sensing instrument is in fact a spectral signature of the underlying materials. Many algorithms attempt to find pure spectral signatures in the image data, called endmembers, and use this information to identify the underlying materials within each pixel. The N-FINDR algorithm is one of the most popular and widely used, despite its high computational complexity when applied to high-dimensional images. However, implementations of this algorithm on reconfigurable hardware are not yet available in the literature, despite the potential advantages that can be gained from processing the hyperspectral images onboard the sensor using specialized hardware devices. In this paper, we present a field programmable gate array design of the algorithm, which has been implemented on a Virtex-4 XC4VFX60 FPGA and evaluated using the well-known "Cuprite" image (a standard benchmark in hyperspectral imaging applications). Our experimental results demonstrate that a hardware version of the N-FINDR algorithm can significantly outperform an equivalent software version and retain excellent pure spectral extraction accuracy.**

I. Introduction

Hyperspectral imaging, also known as imaging spectroscopy, is a technique that has been widely used during recent years in Earth and planetary remote sensing [1]. It generates hundreds of images, corresponding to different wavelength channels, for the same area on the surface of the Earth. The concept of hyperspectral imaging originated at NASA's Jet Propulsion Laboratory in California, which developed instruments such as the Airborne Imaging Spectrometer (AIS), then called AVIRIS (for Airborne Visible Infra-Red Imaging Spectrometer [2]). This system is now able to cover the wavelength region from 400 to 2500 nanometers using 224 spectral channels, at nominal spectral resolution of 10 nanometers. As a result, each pixel (considered as a vector) collected by a hyperspectral instrument can be seen as a *spectral signature* or 'fingerprint' of the underlying materials within the pixel (see Figure 1).

One of the great challenges in remotely sensed hyperspectral image analysis is computational complexity resulting from the need to process enormous data volumes [3]. With recent advances in reconfigurable computing, many image processing algorithms can be accelerated using high-performance FPGAs [4]. One of the fundamental tasks in hyperspectral image processing is endmember extraction which has found many applications in data exploitation, especially spectral unmixing [5]. Over the last years, many algorithms have been developed with the purpose of finding "spectral endmembers" [5], which are assumed to be pure signatures in hyperspectral data sets. Such pure signatures can then be used to estimate the abundance or concentration of materials in mixed pixels, thus allowing sub-pixel analysis of hyperspectral images. The N-FINDR algorithm has been widely used in endmember extraction [6]. This algorithm has a very expensive computational cost, a fact that has generally prevented its exploitation in valid respose times in a wide range of applications, including environmental monitoring, military applications or hazard and threat assessment/tracking. The flexibility, high performance and reduced energy consumption of FPGAs make them particularly attractive in remote sensing applications which require a response in real- or near real-time [4].

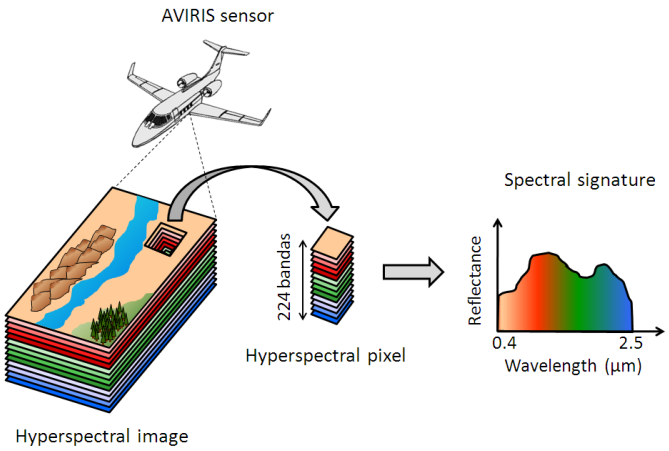


Fig. 1. The concept of hyperspectral imaging.

The remainder of the paper is organized as follows. Section II formulates the spectral unmixing problem in mathematical

terms. Section III describes the original N-FINDR algorithm. Section IV describes its implementation on a Xilinx Virtex-4 XC4VFX60 FPGA. Section V provides an experimental assessment of both endmember extraction accuracy and processing performance of the proposed FPGA-based algorithm, using the well-known "Cuprite" image over the Cuprite mining district in Nevada. Finally, section VI concludes with some remarks and hints at plausible future research lines.

II. Spectral Unmixing

In order to define the spectral unmixing problem in mathematical terms, let us assume that a remotely sensed hyperspectral scene with n bands is denoted by $\mathbf{I}$, in which the pixel at the discrete spatial coordinates (i, j) of the scene is represented by a vector $\mathbf{X}(i,j) = [x_1(i,j), x_2(i,j), \cdots, x_n(i,j)] \in \Re^n$, where $\Re$ denotes the set of real numbers in which the pixel's spectral response $x_k(i,j)$ at sensor channels $k = 1, \ldots, n$ is included. Under the linear mixture model assumption [3] [7], each pixel vector in the original scene can be modeled using the following expression:

$$\mathbf{X}(i,j) = \sum_{z=1}^{p} \Phi_z(i,j) \cdot \mathbf{E}_z + \mathbf{n}(i,j), \tag{1}$$

where $\mathbf{E}_z$ denotes the spectral response of endmember z, $\Phi_z(i,j)$ is a scalar value designating the fractional abundance of the endmember z at the 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 $\{\mathbf{E}_z\}_{z=1}^{p}$ of endmembers and their correspondent abundance fractions $\{\Phi_z(i,j)\}_{z=1}^{p}$ at each pixel $\mathbf{X}(i,j)$. The derivation and validation of the correct suite of endmembers has remained a challenging and goal for the past years (not only in terms of adequate spectral signature extraction [5], but also in terms of computational complexity [8]).

III. N-FINDR algorithm

This algorithm attempts to automatically find the simplex of maximum volume that can be inscribed within the hyperspectral data set. The original N-FINDR algorithm developed by Winter [6] can be summarized as follows:

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).
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:

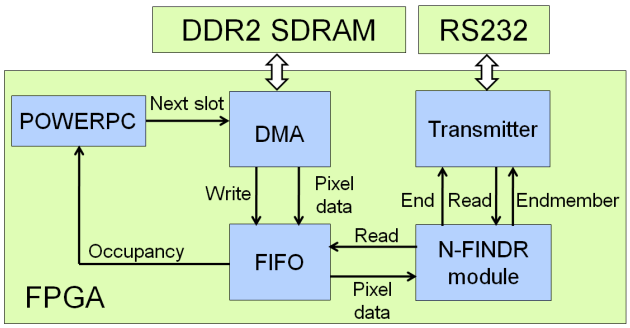


Fig. 2. Hardware architecture of the complete system.

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

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 $V(\mathbf{E}_1^{(k)}, \mathbf{X}(i,j), \cdots, \mathbf{E}_p^{(k)})$, 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 an iterative fashion, using as many iterations as needed until there are no more replacements of endmembers.

In this work, we use the PCA of ENVI 4.0 software to generate dimensionally reduced images to be the input of the N-FINDR algorithm.

IV. Proposed architecture

Figure 2 describes the general architecture of the hardware used to implement the N-FINDR algorithm, along with the I/O communications. For data input, we use a DDR2 SDRAM and a DMA (controlled by a PowerPC using a prefetching approach) with a FIFO to store pixel data. N-FINDR module is used to implement our version of the N-FINDR algorithm. Finally, a transmitter is used to send the endmembers via a RS232 port.

The most time consuming part of the algorithm is the *volume calculation*. The limited available resources in a small or medium FPGA to calculate determinants of large order, makes it difficult to develop an efficient implementation of the algorithm and this is the reason because there are not FPGA implementations of the algorithm in the literature. To calculate determinants, it is advisable to use the fundamental properties of the determinants and apply them systematically

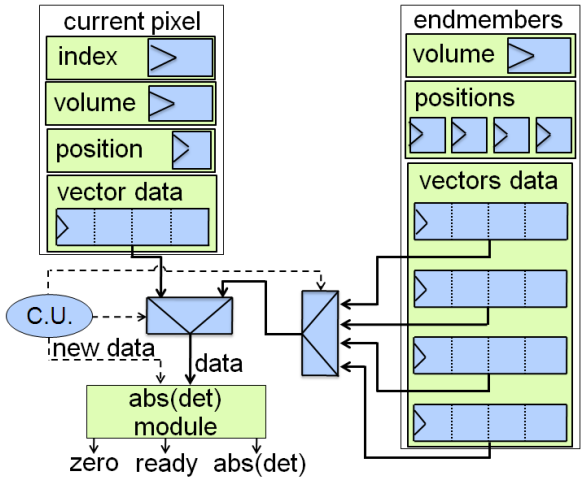


Fig. 3. Hardware architecture to implement the N-FINDR algorithm.

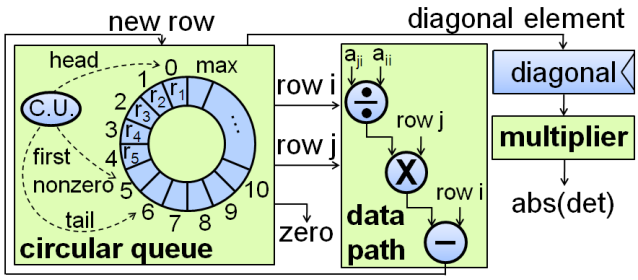


Fig. 4. Hardware architecture of the abs(det) module.

to transform the determinant in others who are increasingly easy to calculate, down to one which is trivial. For the design of the algorithm we use the Gauss elimination method in order to have a triangular matrix.

Figure 3 shows the hardware architecture used to implement the *volume calculation* step. We use registers to store the pixel vectors selected as endmembers until the moment, their positions in the image and their volume, and also the current pixel vector data, his position, his greater volume and the index inside the matrix where it is obtained. Moreover, we have included a module that calculates the absolute value of the determinant using the Gauss elimination method:

First, for $j = 2, ..., n$ we take a multiple a_{j1}/a_{11} of the first row and subtract it to the j-th row, to make $a_{j1} = 0$. Thus, we have knocked out all elements of matrix **A** below the 'pivot' element a_{11} in the first column. Now, for $j = 3, ..., n$, we take a multiple a_{j2}/a_{22} of the second row and subtract it to the j-th row. When we have finished this, all sub-diagonal elements in the second column are zero, and we are ready to process the third column. Applying this process to columns $i = 1, ..., n-1$ completes the matrix triangulation process and matrix **A** has been reduced to upper triangular form. These operations are carried out by the data path (see Figure 3).

Obviously, if one of the diagonal pivots a_{ii} is zero, we cannot use aii to knock out the elements below it; we cannot change a_{ji} by subtracting any multiple of $a_{ii} = 0$ to it. We must switch row i with another row k below it, which contains a nonzero element a_{ki} in the i-th column. Now the new pivot a_{ii} is not zero, and we can continue the matrix triangulation process. If $a_{ki} = 0$ for $k = i, ..., n$, then it will not be satisfactory to switch row i with any of rows below it, as all the potential pivots are zero and therefore det **A** = 0. This behaviour has been implemented using a modified circular queue with a small control unit (see Figure 4).

Finally, the multiplier calculates the multiplication of the main diagonal elements of the triangular matrix and obtains the absolute value.

With this implementation we can extract up to 21 endmembers (the typical number of endmembers per scene lies below this range) with almost total use of available resources of a small FPGA (all embedded DSP48Es multipliers and 97% of the FPGA slices). Table I shows the resources used for our hardware implementation of the proposed N-FINDR algorithm design for different numbers of endmembers to be extracted.

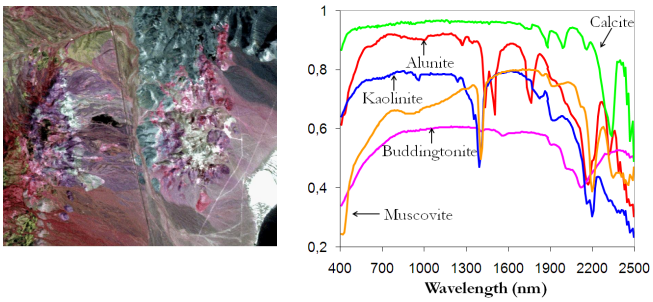
TABLE I
SUMMARY OF RESOURCE UTILIZATION FOR THE FPGA-BASED IMPLEMENTATION OF THE N-FINDR ALGORITHM FOR DIFFERENT NUMBERS OF ENDMEMBERS ON A VIRTEX-4 XC4VFX60

Component	Number of endmembers	Number of DSP48Es	Number of slices	Maximum frequency
N-FINDR module	9	92	6231 (24%)	43.1 MHz
	16	128	11700 (46%)	42.9 MHz
	18	128	14056 (55%)	42.8 MHz
	19	128	17577 (69%)	42.6 MHz
	21	128	24622 (97%)	42.3 MHz
RS232 Transmitter	-	0	71 (0.28%)	208 MHz
DMA Controller	-	0	367 (1.4%)	102 MHz

V. EXPERIMENTAL RESULTS

The hardware architecture described in section IV has been implemented using VHDL language for the specification of the N-FINDR module. Further, we have used the Xilinx ISE environment and the Embedded Development Kit (EDK) environment to specify the complete system. The full system has been implemented on a ML410 board, a low-cost reconfigurable board with a single Virtex-4 XC4VFX60 FPGA component, a DDR2 SDRAM DIMM slot which holds up to 2GBytes, a RS232 port, and some additional components not used by our implementation. We use a Xilinx Virtex-4 XC4VFX60 FPGA because is based on the same architecture as other FPGAs [11] that have been certified by several international agencies for space operation. This FPGA is very close to the space-grade Virtex-4QV XQR4VFX60 FPGA so we could immediately implement our design on it.

The hyperspectral dataset used in these experiments is the well-known Cuprite scene (see Figure 5(a)), collected by NASA Jet Propulsion Laboratory's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS). The scene comprises a relatively large area (350 lines by 350 samples), and 224 spectral bands. The site is well understood mineralogically,



(a) (b)

Fig. 5. (a) False color composition of the AVIRIS hyperspectral over the Cuprite mining district in Nevada. (b) U.S. Geological Survey mineral spectral signatures used for validation purposes.

TABLE II
SPECTRAL ANGLE-BASED SIMILARITY SCORES BETWEEN THE ENDMEMBERS EXTRACTED AND THE SELECTED USGS REFERENCE SIGNATURES

USGS mineral	Alunite	Buddingtonite	Calcite	Kaolinite	Muscovite
SA	0.084	0.073	0.089	0.138	0.092

and has several exposed minerals of interest including alunite, buddingtonite, calcite, kaolinite and muscovite. Reference ground signatures of the above minerals, available in the form of a U.S. Geological Survey library (USGS) will be used to assess endmember signature purity in this work (see Figure 5(b)).

We first conducted an experiment-based cross examination of endmember extraction accuracy to assess the spectral similarity between the USGS library spectra and the corresponding endmembers extracted by the proposed implementation of the N-FINDR algorithm. Table II shows the spectral angle (SA) between the most similar endmembers detected by our FPGA-based implementation.

It should be noted that the SAD between a pixel vector $\mathbf{X}(i,j)$ selected by the N-FINDR and a reference spectral signature $\mathbf{S}_k$ is given by:

$$\text{SA}\left[\mathbf{X}(i,j),\mathbf{S}_k\right] = \cos^{-1}\frac{\mathbf{X}(i,j)\cdot\mathbf{S}_k}{\|\mathbf{X}(i,j)\|\cdot\|\mathbf{S}_k\|}, \qquad (3)$$

where smaller SA values indicate higher spectral similarity. For all the cases our approach obtains excellent results.

We have performed a comparison of our proposed FPGA design with an equivalent software version developed in C language and executed on a PC with AMD Athlon 2.6 GHz processor and 512 Mb of RAM. While the hardware version took 13.46 seconds, the software version needed 502 seconds, so our FPGA implementation shows a speedup of 37.

VI. CONCLUSIONS

The number of remote sensing applications requiring fast response of algorithm analysis has been growing exponentially in recent years. Current sensor design practices can greatly benefit from the inclusion of radiation-hardened FPGAs, which can be easily mounted or embedded in the sensor due to its compact size and low-weight, which does not compromise mission payload. In this paper, we present the experimental results of our FPGA implementation of the N-FINDR algorithm, one of the most well-known approaches for hyperspectral data analysis in the remote sensing community. Our experimental results, conducted on a Virtex-4 XC4VFX60, demonstrate that our architecture can extract endmembers with highly satisfactory spectral purity. Further, our proposed hardware version of the algorithm can significantly outperform (in terms of computation time) an equivalent software version.

As future work, we are investigating FPGA implementations of techniques for estimating the number of endmembers in the scene, such as the virtual dimensionality concept in [12], as well as techniques for estimating endmember abundances in order to provide a full spectral unmixing chain.

ACKNOWLEDGMENT

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 by the projects with references TIN2006-03274, AYA2008-05965-C04-02, AYA2009-13300-C03-02 and TIN2009-09806. Funding from Junta de Extremadura (local government) under project PRI09A110 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. 22, pp. 1147–1153, 1985.

[2] R. O. Green et al., "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment*, vol. 65, pp. 227–248, 1988.

[3] C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Kluwer: New York, 2003.

[4] T. A. El-Ghazawi, E. El-Araby, M. Huang, K. Gaj, V. V. Kindratenko and D. A. Buell, "The promise of high-performance reconfigurable computing," *IEEE Computer*, vol. 41, pp. 69–76, 2008.

[5] 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*, vol. 42, pp. 650–663, 2004.

[6] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," *Proceedings of SPIE*, vol. 3753, pp. 266-275, 1999.

[7] D. Heinz and C.-I. Chang, "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 529-545, 2001.

[8] A. Plaza and C.-I Chang, *High performance computing in remote sensing*. CRC Press: Boca Raton, 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 Transactions on Geoscience and Remote Sensing*, 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] Xilinx. Available online: http://www.xilinx.com/publications/prod_mktg/virtex5qv-product-table.pdf

[12] 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, pp. 608619, 2004.