High Level Methodology To Implement The Vertex Component Analysis Algorithm In A GPU

Ricardo Topham González, Gustavo Marrero Callicó, Sebastián López Suárez

Integrated Systems Design Division, IUMA

Las Palmas de Gran Canaria, Spain

ricardo.topham@gmail.com, gustavo@iuma.ulpgc.es, seblopez@iuma.ulpgc.es

***Abstract*—This paper presents a novel approach towards implementing the Vertex Component Analysis (VCA) endmember extraction algorithm in a GPU, in order to take advantage of the architecture of these devices. The developed high level methodology makes use of MATLAB's Parallel Computing Toolbox and its functions available for programming a GPU.**

Keywords: endmember, GPU, hyperspectral, simplex, VCA.

I. INTRODUCTION

Hyperspectral remote sensing has become increasingly popular over the last years. At the same time, several algorithms for endmember extraction have been published in scientific literature. They all have undergone exhaustive tests and comparisons by authors. Despite the different nature of these algorithms, they all demand an enormous computational effort in order to extract the endmembers of a given hyperspectral image, which combined to the high dimensionality of hyperspectral cubes, seriously jeopardises their use in some environments.

This paper presents a novel approach in order to implement the Vertex Component Analysis (VCA) algorithm [1] in a GPU. The goal is to develop a high level methodology to program and execute it in a GPU taking advantage of its architecture and its potential for parallel computing. For this purpose a NVIDIA GeForce GTX 480 [2] and MATLAB's Parallel Computing Toolbox [3] will be used. The methodology will use several functions available in MATLAB's toolbox built for programming a GPU. In addition, this methodology can be extrapolated easily to other hyperspectral analysis algorithms.

The rest of this paper is organized as follows. Section II describes the original VCA algorithm, while Section III exposes the proposed high level methodology together with an explanation of the change introduced to the original algorithm in order to optimize its execution. Section IV presents the most significant results obtained and, finally, Section V outlines the conclusions extracted from this work.

II. THE VCA ALGORITHM

The VCA algorithm has demonstrated to be a more effective solution than other classical endmember extraction algorithms, such as the pixel purity index (PPI) algorithm [4] or the N-FINDR algorithm [5], in the sense that it provides similar results to the ones provided by these two algorithms but demanding a lower computational effort.

The VCA algorithm is based on the algebraic fact that the endmembers are the vertices of a simplex, being the affine transformation of a simplex also a simplex. VCA uses a positive cone defined by the hyperspectral data to be processed, which projected on a properly chosen hyperplane gives a simplex with vertices corresponding to the endmembers. After projecting the data onto the selected hyperplane, the VCA projects all image pixels to a random direction, obtaining the first endmember as the pixel with the largest projection. The other endmembers are identified by iteratively projecting the data onto a direction orthonormal (given by a vector named f) to the subspace spanned by the endmembers already determined. The new endmember is then selected as the pixel corresponding to the extreme projection, and the procedure is repeated until the whole set of p endmembers is found.

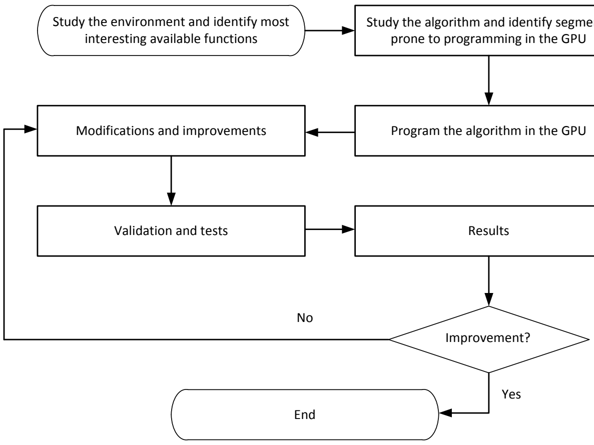
III. HIGH LEVEL METHODOLOGY PROPOSED

The high level methodology proposed in this paper can be split into 8 different stages:

1. Study of the environment and identification of the most interesting available functions.
2. Study of the algorithm and identification of segments prone to programming in the GPU.
3. Programming of the algorithm in the GPU.
4. Modifications and improvements.
5. Validation and tests.
6. Results.
7. Improvements?
8. End.

It is important to point out that the 7^{th} stage is crucial as a decision has to be made: if results improve, then the implementation is ready, if results do not improve, it might be necessary to go back to stage 4 in order consider other possible solutions. The developed methodology is shown in FIGURE I.

FIGURE I. HIGH LEVEL METHODOLOGY PROPOSED TO IMPLEMENT THE VCA ALGORITHM IN A GPU



As can be seen, this methodology can be easily used with other hyperspectral analysis algorithms.

One modification has been taken into account in the process of programming the VCA algorithm in a GPU. In an effort to reduce execution time, the normalisation operation was written off as. Tests confirmed that, as expected, that operation could be saved without affecting results.

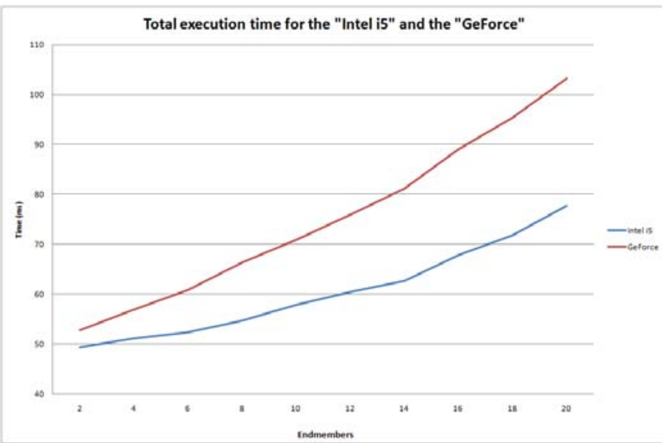
IV. RESULTS

In this section the performance of the optimised VCA is compared in an Intel i5 650 CPU and an NVIDIA GeForce GTX 480.

Well known hyperspectral images represent an excellent test bench for the purpose of comparing both algorithms, since the signature endmembers as well as their fractional abundances are known in advance. In particular, the testbench tool used in this work, the *demo_vca* software, is available in the VCA's authors webpage [6].

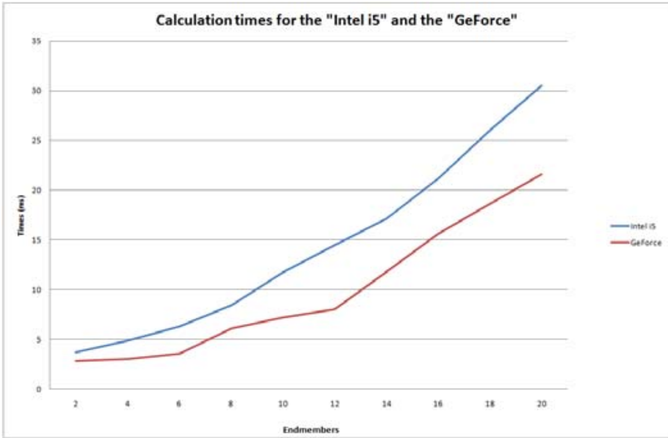
The first test considers the total execution times in both devices, which is shown in FIGURE II.

FIGURE II. TOTAL EXECUTION TIME.



The execution time in the GPU is greater than that of the CPU. The reason for this difference are the additional memory transfers that are necessary to bring the data to the GPU's memory. The real performance of both systems can be compared when only calculation times are taken into account, as shown in FIGURE III.

FIGURE III. CALCULATION TIMES.



V. CONCLUSION

As it can be seen in the last graph, the power to carry out intense computations of the GPU exceeds the one of the CPU. The main drawback is that memory access has to be carefully studied as it is the main bottleneck in these types of algorithms. In any case, regarding the VCA algorithm, differences are in the order of milliseconds, proving the suitability of the proposed methodology.

In addition, an optimisation of the original VCA algorithm was performed, reducing total execution times, whether doing so in a CPU or in a GPU. With the modification presented in this paper, the VCA is implemented in a more efficient way.

REFERENCES

[1] J. Nascimento, J. Bioucas (2005). "Vertex component analysis: a fast algorithm to unmix hyperspectral data", IEEE Transactions on Geoscience and Remote Sensing, v. 43, nº. 4, pp. 898-910.

[2] NVIDIA GeForce GTX 480/470/465 GPU Datasheet (2010).

[3] Parallel Computing Toolbox™ 5 User's Guide, The MathWorks Inc. (2010).

[4] F. Chaudhry, C. Wu, W. Liu, C. Chang, A. Plaza (2006). "Pixel purity index-based algorithms for endmember extraction from hyperspectral imagery". Transworld Research Network, pp.30-62.

[5] M. Zortea, A. Plaza (2009). "A quantitative and comparative analysis of different implementations of N-FINDR: A fast endmember extraction algorithm", IEEE Geoscience and Remote Sensing Letters, v. 6, nº. 4, pp. 787-791.

[6] http://www.lx.it.pt/~bioucas/code.htm. Last visited: 20/07/11.