# Hardware Implementation of Vertex Component Analysis algorithm

Pablo Horstrand Andaluz, Sebastián López Suárez, Roberto Sarmiento Rodríguez
Integrated System Design Division, IUMA
Las Palmas de Gran Canaria, Spain
pablohorstrand@gmail.com, seblopez@iuma.ulpgc.es, roberto@iuma.ulpgc.es

***Abstract*—This paper presents various modifications on the endmember extraction algorithm Vertex Component Analysis (VCA), in order to facilitate the task of implementing it in hardware as well as improving the performance of execution. The results of both implementations are very similar as it will be shown in this work, proving that the modifications still perform well the algorithm with a less complexity and computational cost.**

***Keywords: hyperespectral image, endmember, simplex, VCA.***

## I. Introduction

During the last years, several algorithms for endmember extraction have been published in the scientific literature, which have been exhaustively tested and compared among them in other recent works. 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 jeopardize their use in application domains under real time constraints.

In this sense, this paper introduces a modified version of one of the most popular endmember extraction algorithms, the vertex component analysis (VCA) algorithm [1]. The proposed modified vertex component analysis (MVCA) algorithm is able to reproduce the results of its original counterpart, while exhibiting a much lower computational complexity. This is mainly accomplished by introducing a low complexity orthogonalization method, inspired by the Gram–Schmidt algorithm, in order to compute an orthogonal vector to the subspace spanned by the endmembers that have been already determined at each iteration of the VCA algorithm, as well as by utilizing integer instead of floating-point arithmetic when projecting the hyperspectral data onto the direction determined by the orthogonal vector previously mentioned.

The rest of this paper is organized as follows. Section II describes the original VCA algorithm, while Section III exposes the proposed MVCA algorithm together with an explanation of the changes introduced with respect to the VCA algorithm. 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 [2] or the N-FINDR algorithm [3], 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. The Modified VCA (MVCA) Algorithm

Three modifications have been introduced in the process of computing the vector $f$. First, the norm of this vector is not obliged to be equal to 1, which means that the operations performed in the VCA algorithm in order to normalize vector $f$ are saved in the proposed MVCA algorithm. The second modification consists in fixing the vector $w$ to $w$ = [1,1,…,1] rather than generating a random vector at each iteration. The third and more important modification in terms of computational cost savings is based on changing the mechanism adopted in the VCA algorithm for the calculation of a vector orthogonal to the subspace spanned by the endmembers that have been already determined. In particular, this work proposes to compute $f$ by first obtaining an orthogonal set of $i$ vectors $U = \{u_1, u_2, \ldots, u_i\}$ from the set $E = \{e_1, e_2, \ldots, e_i\}$ defined by the $i$ endmembers that have been already computed by following the Gram–Schmidt orthogonalization algorithm:

$$\boldsymbol{u_k} = \boldsymbol{e_k} - \sum_{j=1}^{k-1} \boldsymbol{proj}(\boldsymbol{e_k}, \boldsymbol{u_j}); \quad \{\boldsymbol{k = 1\ to\ i\ and\ u_1 = e_1}\} \tag{2}$$

, where *proj* stands for the projection operator, defined as:

$$\boldsymbol{proj}(\boldsymbol{e_k}, \boldsymbol{u_j}) = \frac{<\boldsymbol{e_k}, \boldsymbol{u_j}>}{<\boldsymbol{u_j}, \boldsymbol{u_j}>}\boldsymbol{u_j} \tag{3}$$

Once $U$ has been obtained, $f$ is computed in the MVCA algorithm as follows:

$$\boldsymbol{f} = \boldsymbol{w} - \sum_{l=1}^{i} \boldsymbol{proj}(\boldsymbol{w}, \boldsymbol{u_l}) \tag{4}$$

which assures that the vector $f$ is orthogonal to the subspace spanned by the endmembers contained in $E$. As it will be detailed in the next section of this letter, this change allows to drastically reducing the number of flops required for the computation of the $f$ vector when compared with the original VCA algorithm.

Once $f$ has been computed, the hyperspectral image $Y$ must be projected onto the direction indicated by this vector. In order to further reduce the computational complexity of the proposed MVCA algorithm, this projection is performed using integer rather than floating point arithmetic, based on the idea that this modification should not alter the position of the projection extreme (although the value of the projection itself will definitively change). For this purpose, the floating point numbers contained in $Y$ and $f$, that we will assume compliant with the IEEE 754-2008 standard [4], should be converted to integers.

## IV. Results

In this section, the performance of the VCA and the MVCA algorithms are compared.

Artificial 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 hyperspectral images used in this work were generated by the *demo_vca* software tool available at [5].

In order to evaluate the accuracy of both algorithms, the spectral angle $\theta_i$ between the $i$-th extracted endmember $e_i$ and its correspondent real spectral signature $e\_r_i$ is calculated as follows:

$$\boldsymbol{\theta_i} = \mathbf{arccos}\frac{<\boldsymbol{e_i}, \boldsymbol{e\_r_i}>}{||\boldsymbol{e_i}||\,||\boldsymbol{e\_r_i}||} \tag{5}$$

Once the $p$ spectral angles have been computed, the spectral root mean square error (RMSE) $\theta_{RMSE}$ is calculated.

Table I tabulates the average difference ($dif\_\theta_{avgRMSE} = VCA\_\theta_{avgRMSE} - MVCA\_\theta_{avgRMSE}$) between both spectral root mean square errors for different values of *SNR* and *p*. Positive values of $dif\_\theta_{avgRMSE}$ in Table I indicate that the MVCA provides a more accurate endmember extraction than the VCA algorithm and vice versa.

TABLE I
AVERAGE DIFFERENCE VALUES (IN DEGREES) BETWEEN THE VCA AND THE MVCA ALGORITHMS

| | | SNR | | | | |
|---|---|---|---|---|---|---|
| | | 10dB | 20dB | 30dB | 40dB | 50dB |
| p | 3 | -0.993 | 0.013 | -0.012 | -0.012 | 0 |
| | 4 | 0.018 | -0.525 | -0.052 | -0.009 | -0.010 |
| | 5 | -0.146 | 0.874 | 0.042 | -0.135 | 0.006 |
| | 6 | 1.147 | -0.496 | -0.064 | 0.424 | 0.012 |
| | 7 | 0.441 | -1.190 | 0.396 | -0.047 | 0.119 |
| | 8 | -0.710 | -0.602 | -0.128 | 0.059 | 0.015 |
| | 9 | -0.491 | -1.083 | -0.079 | 0.110 | -0.003 |
| | 10 | 2,271 | 0.022 | 0.073 | -0.010 | -0.056 |

## V. Conclusion

As it is seen from this table, there are some cases where the MVCA algorithm performs better than the VCA algorithm and others where the opposite occurs. In any case, all the differences, positive or negative, are extremely small which proves the suitability of the proposed MVCA algorithm for extracting a variable set of $p$ endmembers in hyperspectral images with different levels of noise.

The complexity of the proposed algorithm is much lower than the original VCA, thus all the modifications presented in this paper allow implementing the VCA algorithm 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] 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.

[3] 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.

[4] ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic, 1985.

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