FPGA IMPLEMENTATION OF A MAXIMUM VOLUME ALGORITHM FOR ENDMEMBER EXTRACTION FROM HYPERSPECTRAL IMAGERY

Cong Li[1,2], *Lianru Gao*[1,*], *Antonio Plaza*[3], *Bing Zhang*[1]

1. Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, 100094, China
2. University of Chinese Academy of Sciences, Beijing, 100049, China
3. Department of Technology of Computers and Communications, University of Extremadura, 10071 Cáceres, Spain

ABSTRACT

Endmember extraction is a very important technique in hyperspectral unmixing. Due to the fact that more applications require real or near real-time processing capabilities, high performance computing based on field-programmable gate array (FPGA) for endmember extraction has received considerable interest in recent years. In this paper, we propose a real-time implementation of maximum volume algorithms (RT-MaxV) using a Kintex-7 FPGA. The proposed RT-MaxV does not need dimensionality reduction like other real-time fast simplex growing algorithms (RT-FSGA). Our experimental results, obtained using the AVIRIS Cuprite data set, indicate that the proposed algorithm has better accuracy and performance than RT-FSGA, and its FPGA implementation achieves real-time processing capability in the considered problem.

Index Terms— Endmember extraction, real-time maximum volume algorithm (RT-MaxV), field-programmable gate array (FPGA).

1. INTRODUCTION

Hyperspectral unmixing has been widely studied due to the existence of mixed pixels in hyperspectral images. One of the key steps in spectral unmixing is endmember extraction, which aims at extracting the spectral signatures of pure materials from hyperspectral images [1].

Many state-of-the-art algorithms have been developed and proved effective to solve this problem. For instance, N-FINDR [2] and simplex growing algorithm (SGA) [3] are two typical methods which both aim at finding the maximum volume formed by endmembers in spectral space. The SGA has been implemented in FPGAs (RT-FSGA) for real-time performance in [4] in order to meet the requirements of missions subject to time constraints.

These two algorithms have their own strengths and weaknesses. N-FINDR can extract endmembers automatically with a relatively good accuracy, but in its initialization stage, the initial set of pixels is generated randomly which could make the extraction result inconsistent among different iterations. Another drawback of N-FINDR is that it needs to traverse all the possible combinations of pixels in the image considered as endmembers, so the amount of computations required is very large, and this computational complexity increases dramatically with an increase in the number of endmembers. On the other hand, the SGA extracts one endmember in each loop, which is not an exhaustive search but saves a lot of time. However, the SGA still exhibits the same drawback as N-FINDR: for calculating the determinant associated to the volume computation, the matrix formed by the set of selected pixels must be a square matrix. This means that a previous dimensionality reduction step must be accomplished before the main loop of the algorithm, which takes significant time and computing resources and leads to loss of information while removing most of the spectral bands out of the data set.

In this paper, a new real-time maximum volume algorithm (RT-MaxV) is proposed to overcome the aforementioned limitations. The RT-MaxV has two important advantages: 1) it takes all information provided by the full set of spectral bands in the original data, namely, it does not require a dimensionality reduction; 2) its computational complexity is amenable for real-time processing. In this paper, the RT-MaxV is implemented on a Kintex-7 FPGA and tested in the task of extracting endmembers from a real hyperspectral image, collected by the AVIRIS instrument over the Cuprite mining district in Nevada. The obtained results show that the proposed algorithm exhibits better performance than both N-FINDR and RT-FSGA.

2. REAL-TIME MAXV (RT-MAXV)

The maximum volume algorithm (MaxV) was first proposed by Geng et al [5], [6]. It is based on a new volume formula for the simplex. The volume of a simplex whose p vertices are $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p$ can be calculated as follows:

$$V(\mathbf{W}) = \frac{1}{(p-1)!}\sqrt{|\det(\mathbf{W}^{\mathrm{T}}\mathbf{W})|} \qquad (1)$$

*Corresponding author: gaolr@radi.ac.cn

where $\mathbf{W} = [\mathbf{e}_2 - \mathbf{e}_1, \mathbf{e}_3 - \mathbf{e}_1, ..., \mathbf{e}_p - \mathbf{e}_1]$. Due to the fact that $\mathbf{W}^T\mathbf{W}$ is a square matrix in any case, no dimensionality reduction is needed before endmember extraction for a hyperspectral data set with L bands.

The MaxV algorithm also takes advantage of the simplex growing method in [3]. It first finds the initial endmember and then extracts one endmember in each outer-loop. MaxV can be summarized by the following steps:

Step 1: Initialization:

Choose the two pixels with the longest distance to be the first and second endmember, $\mathbf{e}_1$ and $\mathbf{e}_2$, and let $i = 3$.

Step 2: Volume calculation and comparison:

In the inner-loop, for each pixel $\mathbf{r}$, calculate the volume $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r})$ defined by (1). A variable *det_max* is used to store the largest determinant within this inner-loop. The pixel that provides the largest $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r})$ is extracted as endmember $\mathbf{e}_i$.

Step 3: In the outer-loop, if $i \le p$, then make $i = i + 1$ and go to step 2. Otherwise, the subset $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$ is the set of desired p endmembers.

At this point, it is important to reiterate that MaxV does not need any dimensionality reduction and that it effectively exploits all the spectral bands in the original hyperspectral image. However, calculating the determinant of each matrix still requires significant computational complexity, and this operation is not suitable for being implemented on FPGAs. To address this issue, we use Woodbury's identity [7] as follows:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \quad (2)$$

$$\det(\mathbf{A} + \mathbf{u}\mathbf{v}^T) = (1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u})\det(\mathbf{A}) \quad (3)$$

where $\mathbf{A}$ is the original square matrix, and $\mathbf{u}\mathbf{v}^T$ is the outer product of two vectors $\mathbf{u}$ and $\mathbf{v}$. Then $\mathbf{A} + \mathbf{u}\mathbf{v}^T$ is the new matrix that replaces one row or one column.

Let $\mathbf{W} = [\mathbf{e}_2 - \mathbf{e}_1, \mathbf{e}_3 - \mathbf{e}_1, ..., \mathbf{e}_j - \mathbf{e}_1]$, and $\mathbf{A} = \begin{pmatrix} \mathbf{W}^T\mathbf{W} & \mathbf{o} \\ \mathbf{o} & 1 \end{pmatrix}$

in which $\mathbf{o}$ is the zero vector. When a new pixel $\mathbf{r}$ is updated, the last column and the last row of $\mathbf{A}$ should be replaced, which means that the inverse matrix and determinant can be updated by using Woodbury's identity twice in each iteration. The vector definitions are given by:

$$\mathbf{v}_1 = \mathbf{u}_2 = [0, ..., 0, 1] \quad (4)$$

$$\mathbf{u}_1 = [(\mathbf{e}_2 - \mathbf{e}_1)^T(\mathbf{r} - \mathbf{e}_1), \ldots, (\mathbf{r} - \mathbf{e}_1)^T(\mathbf{r} - \mathbf{e}_1)]^T \quad (5)$$

$$\mathbf{v}_2 = [(\mathbf{r} - \mathbf{e}_1)^T(\mathbf{e}_2 - \mathbf{e}_1), \ldots, (\mathbf{r} - \mathbf{e}_1)^T(\mathbf{e}_j - \mathbf{e}_1), -1]^T \quad (6)$$

and the final matrix is given by the following term:

$$\mathbf{A} + \mathbf{u}_1\mathbf{v}_1^T + \mathbf{u}_2\mathbf{v}_2^T = [\mathbf{e}_2 - \mathbf{e}_1, ..., \mathbf{e}_j - \mathbf{e}_1, \mathbf{r} - \mathbf{e}_1]^T[\mathbf{e}_2 - \mathbf{e}_1, ..., \mathbf{e}_j - \mathbf{e}_1, \mathbf{r} - \mathbf{e}_1] \quad (7)$$

It also should be noted that all the determinants in an inner-loop are updated from the same initial value *det_old*, which equals to *det_max* in the last outer-loop. Instead of updating $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r}_i)$ from $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r}_{i-1})$, the former computed results will not influence the latter results within the inner-loop. In this way, the accumulated error can be significantly reduced. Another improvement is the initialization performed in step 1. As reported in [4], the first endmember can be generated by finding the pixel vector with maximum length. The new method proposed in this paper is called RT-MaxV, which can achieve real-time processing capability as shown in subsequent sections.

3. FPGA IMPLEMENTATION OF RT-MAXV

The hardware architecture of RT-MaxV on FPGA is shown in Fig. 1. The image data is stored in a SDRAM, and pixel $\mathbf{r}$ is read by a determinant calculation module, which gets the determinant *det_new*. The compare and update module is used to compare the new generated determinant *det_new* and the former maximum *det_max*. If *det_new* > *det_max*, then we let *det_max* = *det_new* and update the corresponding inverse matrix and pixel coordinates.

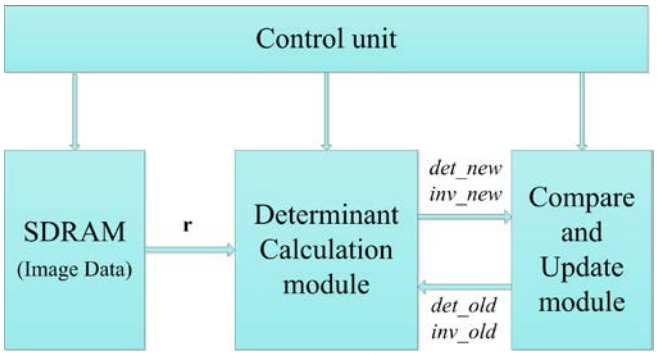


Figure 1. Hardware structure of RT-MaxV on FPGA.

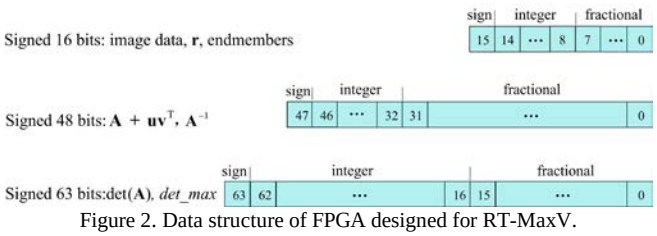


Figure 2. Data structure of FPGA designed for RT-MaxV.

The key part of our FPGA design is the determinant calculation module. To reduce the resources and computing time, the data size in this module should be as small as possible with the premise of ensuring accuracy. By

multiplying $\mathbf{W}$ by its transpose, every element in $\mathbf{W}^T\mathbf{W}$ and its determinant can be very big, and the elements in its inverse matrix may be extremely small at the same time. The data size must find a proper balance, so we enlarged the image data from the range of 0~1 to 0~5, and obtained the fixed-point format shown in Fig. 2.

Fig. 3 shows the data flow of the determinant calculation module. Woodbury's identity should be used twice in each iteration, so that the inputs of the second round should be the determinant and inverse matrix calculated by the first round.

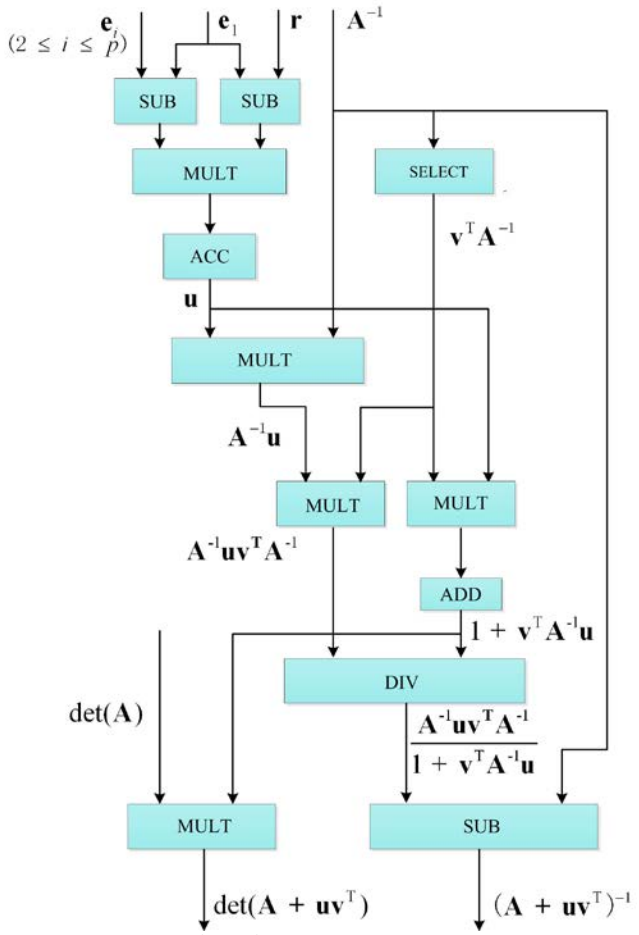
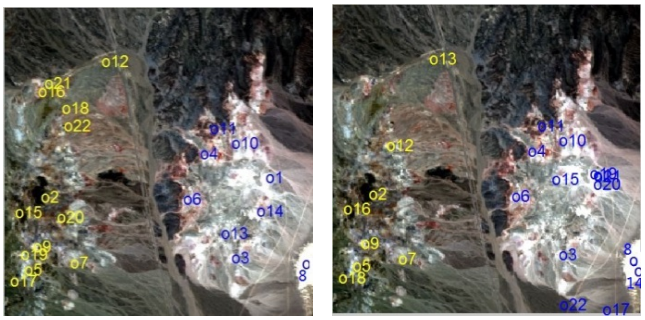


Figure 3. Data flow of the determinant calculation module.

The RT-MaxV has been implemented on a Kintex-7 FPGA KC705 Evaluation Kit of Xilinx. The XC7K325T of Kintex-7 series has 50950 slices, 326080 logic cells and 407600 CLB flip-flops available, and also a total block RAM of 16020 kb. The Kintex-7 FPGA family provides an ideal balance in terms of integration, price, performance, and power consumption, and can double the price-performance with half the power consumption as compared with the vertex-6 family. We use ISE 14.2 as the developing environment, and modelsim SE 10.1a to perform simulation.

4. EXPERIMENTAL RESULTS AND ANALYSIS

The image used in experiments was collected by AVIRIS over the Cuprite mining district in 1997. The data have 224 bands and the size is 350×350 pixels. According to [4], bands 1-3, 105-115, and 150-170 have been removed from the data set because of water absorption and low SNR, retaining 189 spectral bands. There are 5 representative pure minerals as a priori in this image: *alunite*, *buddingtonite*, *calcite*, *kaolinite*, and *muscovite*. These minerals' spectral signatures can be found in a USGS spectral library. The virtual dimensionality (VD) is used to estimate the number of endmembers in this image, which is 22.


(a) (b)
Figure 4. Endmembers extracted by RT-MaxV. (a) Matlab. (b) FPGA.

We have conducted experiments using Matlab and our FPGA implementation of RT-MaxV. Fig. 4(a) and 4(b) respectively show the endmembers extracted by RT-MaxV using Matlab and our FPGA implementation. The endmembers were marked in the same order as they were extracted. For the first 11 endmembers, the FPGA design got exactly the same result as Matlab did, and then there were some differences after that. Although the reflectance values are coded using 16 bits, it is unavoidable that a part of the fractions may be cut off, and this little difference becomes more relevant, to the extent that it may influence the final result after several iterations. If we increased the number of bits used to represent the values during processing, this calculation error could be controlled to be small enough to get the same result as MaxV (at the cost of using more computing resources in the FPGA).

To evaluate endmember extraction accuracy, we used the spectral angle (SA) scores and root mean square error (RMSE), where MaxV, N-FINDR, and SGA are all implemented in Matlab and RT-MaxV is implemented on the FPGA. The results are shown in Table I. These four algorithms get very similar results in terms of SA. N-FINDR and MaxV get much lower RMSEs than SGA. N-FINDR exhibits good accuracy because it traverses all the combinations of pixels, but this takes a very long time. MaxV uses a simplex growing method, so MaxV exhibits

higher computational efficiency than N-FINDR. In general, MaxV outperforms SGA and N-FINDR if we take both accuracy and computational efficiency into consideration. RT-MaxV is the FPGA implemented version of MaxV, and also keeps better balance between accuracy and computing performance than N-FINDR and SGA. It should be noted that RT-MaxV and MaxV should get the same result without the accumulated computing error aforementioned.

Table I
Spectral angle scores (in degrees) between USGS mineral spectra and their corresponding estimated endmembers, and the RMSE after reconstructing the Cuprite scene.

Algorithms	Alunite	Budding-tonite	Calcite	Kaolinite	Muscovite	RMSE
N-FINDR	3.913	4.412	**5.271**	**5.220**	4.120	0.0082
SGA	**3.638**	4.263	5.306	5.323	4.240	0.0189
MaxV	4.286	**3.879**	5.306	5.323	4.240	**0.0077**
RT-MaxV	**3.638**	**3.879**	5.306	5.323	**3.747**	0.0083

Table II shows the computing results of RT-MaxV and RT-FSGA, both on Matlab and the FPGA. The experimental results of RT-FSGA implemented on a FPGA are reproduced from [4]. For RT-MaxV, Matlab used 163.38s to get the result, and our FPGA implementation used 2.17×10^8 clock periods. If clock frequency is set to the system frequency of Kintex-7, namely 200 MHz, then the processing time is 1.09s. Compared to 0.16s of RT-FSGA, the implementation of RT-MaxV is a bit slower. However, as analyzed above, RT-MaxV exhibits better accuracy than RT-FSGA in terms of endmember extraction. Moreover, RT-MaxV does not need dimensionality reduction, as opposed to RT-FSGA.

The newest generation of AVIRIS from National Aeronautics and Space Administration (NASA) is characterized by a sensor data acquisition rate of 17MB/s. The data size of experimental image in this paper is about 55MB, so the AVIRIS sensor requires at least 3.23s to acquire an image with the same size. Our FPGA implementation can complete the endmember extraction in 1.09s, so it strictly meets the real-time requirements of sensors such as AVIRIS new generation. In addition, it should also be noted that our FPGA implementation was not fully optimized yet; this will be the subject of our future developments.

Table II
Processing time of RT-MaxV for Matlab and FPGA implementation.

Implementations	Matlab (s)	FPGA	
		Processing time (s)	clock periods
RT-FSGA	46.58	0.16	3.11×10^7
RT-MaxV	163.38	1.09	2.17×10^8

5. CONCLUSIONS AND FUTURE LINES

Endmember extraction has played a very important role in hyperspectral image analysis. However, high spectral resolution leads to high computational complexity, so there is always an urgent need to realize real or near real-time implementations. In this paper, we have presented an FPGA implementation of a real-time maximum volume algorithm (RT-MaxV) for endmember extraction. This method needs no dimensionality reduction, so it can take full advantage of the information contained in the original spectral signatures without losing any detail. The simplex growing method and a fast matrix determinant computation helped to decrease the computational complexity. The RT-MaxV has better accuracy than other FPGA implementations such as RT-FSGA (based on simplex growing), and also has the advantage that it does not need dimensionality reduction. Our future work is focused on finding a balance among accuracy, speed, and computing resources for the proposed algorithms. For images of larger size, methods like generic endmember re-extraction operation (GERO) can be used.

6. ACKNOWLEDGMENT

This research was supported by the Key Research Program of the Chinese Academy of Sciences under Grant No. KZZD-EW-TZ-18, and by the National High Technology Research and Development Program of China (863) under Grant No. 2011AA120203.

7. REFERENCES

[1] J. M. B. Dias, A. Plaza, N. Dobigeon, M. Parente, etc, "Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.5, no.2, pp.354-379, April 2012.

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

[3] C. I. Chang, C. Wu, W. Liu, and Y. C. Ouyang, "A growing method for simplex-based endmember extraction algorithms," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 44, no. 10, pp. 2804–2819, October 2006.

[4] C. I. Chang, W. Xiong, and C. C. Wu, "Field-programmable gate array design of implementing simplex growing algorithm for hyperspectral endmember extraction," *IEEE Transaction on Geoscience and Remote Sensing*, vol.51, no.3, pp.1693-1700, March 2013.

[5] X. R. Geng, "Target Detection and Classification for Hyperspectral Imagery", Ph.D. dissertation, Institute of Remote Sensing Applications Chinese Academy of Science, Beijing, China, 2005.

[6] X. R. Geng, Y. C. Zhao, F. X. Wang, and P. Gong, "A new formula for a simplex and its application to endmember extraction for hyperspectral image analysis," *International Journal of Remote Sensing*, vol.31, no.4,pp.1027-1035, February 2010.

[7] W. Xiong, C. C. Wu, C. I. Chang, K. Kapalkis, and H. M. Chen, "Fast algorithms to implement N-FINDR for hyperspectral endmember extraction," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 3,pp. 545–564, September 2011.