J Real-Time Image Proc (2019) 16:1681–1694
DOI 10.1007/s11554-017-0679-2

ORIGINAL RESEARCH PAPER

FPGA implementation of a maximum simplex volume algorithm for endmember extraction from remotely sensed hyperspectral images

Cong Li[1] · Lianru Gao[1] · Antonio Plaza[2] · Bing Zhang[1]

Received: 12 September 2016 / Accepted: 25 February 2017 / Published online: 7 March 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract Spectral unmixing is a very important technique for remotely sensed hyperspectral unmixing. Since more hyperspectral applications now require real or near real-time processing capabilities, fast spectral unmixing using field-programmable gate arrays (FPGAs) has received considerable interest in recent years. FPGAs can provide onboard, high computing performance at low power consumption. Another important characteristic of FPGA-based systems is reconfigurability, which makes them more flexible to process different kind of scenes. Pure signature (endmember) extraction is a fundamental step in spectral unmixing, which has been tackled using the maximum volume principle by several algorithms, most notably N-FINDR and simplex growing algorithm (SGA). These algorithms find out the simplex with maximum volume as a mechanism to extract endmembers. However, a previous dimensionality reduction step is generally required, which introduces information loss and additional computational burden. To address these issues, in this work we introduce a new volume calculation formula and further develop a new real-time implementation of a maximum simplex volume algorithm (called RT-MSVA). The proposed RT-MSVA does not need dimensionality reduction, so all spectral bands can be used without losing any information to ensure robust endmember extraction accuracy. Experiments with synthetic and real hyperspectral images have been conducted to evaluate the accuracy and computational performance of our proposed method. Our experimental results indicate that proposed FPGA-based implementation significantly outperforms the corresponding software version and achieves real-time processing performance in the considered problem. It also exhibits better endmember extraction accuracy and comparable performance to other available techniques, such as a real-time implementation of a simplex growing algorithm (RT-FSGA).

Keywords Hyperspectral imaging · Endmember extraction · Field-programmable gate array (FPGA) · Real-time maximum simplex volume algorithm (RT-MSVA)

✉ Lianru Gao
gaolr@radi.ac.cn

Cong Li
licong@radi.ac.cn

Antonio Plaza
aplaza@unex.es

Bing Zhang
zb@radi.ac.cn

[1] Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China

[2] Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politecnica de Cáceres, University of Extremadura, Cáceres, Spain

1 Introduction

Hyperspectral imaging has been widely used in remote sensing during recent years [1–3]. It provides a data cube containing hundreds of spectral bands for a single scene, containing a wealth of information that cannot be captured by other sensors such as multispectral instruments. Hyperspectral unmixing is a key technique in hyperspectral data exploitation and has been widely studied due to the common existence of mixed pixels in hyperspectral images [4, 5]. The linear mixture model (LMM) is frequently adopted to describe mixed pixels in hyperspectral scenes, assuming that the spectrum of each mixed pixel can be

expressed as a linear combination of several endmembers [6]. LMM uses the simplest possible way to describe mixed pixels, ignoring multiple scattering effects between endmembers. The linear mixture model can be described as follows:

$$\mathbf{X} = \mathbf{E} \times \mathbf{A} + \mathbf{W} \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is a hyperspectral image with a total of n pixels, $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p]$ is the endmember matrix, $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ is the abundance matrix, and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$ is an error matrix.

The hyperspectral processing chain can be summarized into three main steps: dimensionality reduction, endmember extraction, and abundance estimation. Dimensionality reduction is conducted to reduce data volume and is implicitly used also to estimate the number of endmembers. However, this step has some drawbacks as discussed later. How to extract a proper set of pure spectral signatures (endmembers) from hyperspectral images is widely regarded as the key task in LMM [7].

There are three main approaches for endmember extraction: geometrical, statistical, and sparse regression based [4]. Geometrical methods are based on the assumption that the mixed pixels lie inside of a simplex in which the corners are the endmembers. In other words, the endmembers are assumed to be on the vertices of the data simplex. Many state-of-the-art algorithms have been developed and proved effective to solve this problem, including pixel purity index (PPI) [8], vertex component analysis (VCA) [9], iterative error analysis (IEA) [10], N-FINDR [11], and simplex growing algorithm (SGA) [12]. PPI projects the data pixels onto a large set of random skewers and counts number of pixels resulting in extreme projections. Those pixels with the highest scores are considered to be the endmembers. This algorithm has low computational complexity, but needs human intervention. In turn, VCA projects each pixel to the hyperplane formed by the previously extracted endmembers and aims to find the largest projection. IEA minimizes the reconstruction error by iteratively extracting endmembers as those pixels that result in maximum error values.

N-FINDR and simplex growing algorithm (SGA) aim at finding the endmembers by calculating the maximum volume formed by pixels in the original hyperspectral image. These two algorithms have their own strengths and weaknesses. N-FINDR can extract endmembers automatically with good accuracy but, in its initialization stage, the initial set of pixels is generated randomly which could make the extraction result inconsistent among different algorithm runs. Another drawback of N-FINDR is that it needs to traverse all the possible combinations of pixels in the image (considered as endmembers). As a result, the computational complexity is high and increases dramatically with an increase in the number of endmembers. On the other hand, the SGA extracts one endmember in each iteration, which is not an exhaustive search but saves a lot of time. However, the SGA shares a drawback with N-FINDR: for calculating the determinant associated with 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 leading to a loss of information while removing most of the spectral bands of the data set. Some recent works have specifically focused on the aforementioned issues. An algorithm using a new determinant calculation formula was proposed in [13, 14] which is called maximum simplex volume algorithm (MSVA). A method to simplify the MSVA algorithm was presented in [15], which reduces the computational complexity and makes it amenable for real-time applications.

An important challenge for hyperspectral image analysis is the processing speed. The fast development of new sensors leads to improved spectral, spatial, and temporal resolutions, together with increased data volumes [16]. The resolution increase provides more detailed information, but demands fast computing solutions at the same time. Hardware accelerators play an important role in high-performance computing for remote sensing. Most notably, clusters [17, 18] and specialized hardware devices such as commodity graphics processing units (GPUs) [19] and field-programmable gate arrays (FPGAs) [20–22] have been used for accelerating hyperspectral computations. There is some excellent work with GPUs on hyperspectral unmixing in [23, 24]. Both GPUs and FPGAs provide highly desirable features such as low weight and high computing performance, but FPGAs also offers reconfigurability and lower energy consumption ratios, in addition to radiation tolerance that still cannot be achieved by GPUs [25].

An FPGA is an integrated circuit whose logic blocks can be defined and configured by users repeatedly. That means that FPGAs offer the possibility of adaptively selecting a proper hyperspectral algorithm to be applied from the control station on Earth anytime [26]. FPGAs also exhibit an architecture that is suitable to implement pipeline processing [27]. FPGA-based hyperspectral image processing has aroused wide interest in the hyperspectral image community, and recent works have explored FPGA-based implementations of endmember extraction algorithms. For instance, the N-FINDR algorithm has been implemented on FPGAs in [28, 29]. The SGA has an improved version (RT-

FSGA) that has also been implemented on FPGAs for real-time performance in [30] in order to meet the requirements of hyperspectral missions subject to near real-time constraints.

In this paper, we develop a new real-time implementation of MSVA (called RT-MSVA), which intends to overcome the aforementioned limitations of maximum volume algorithms such as N-FINDR and SGA. The proposed RT-MSVA has three important advantages: (1) it exploits all the information provided by the full set of spectral bands in the original data, without a prior dimensionality reduction; (2) its computational complexity is amenable for strictly real-time processing; (3) it is suitable for being implemented on FPGAs. The newly developed RT-MSVA has been implemented on a Kintex-7 FPGA and tested using both synthetic and real hyperspectral images. The obtained results show that the proposed algorithm exhibits better performance than both N-FINDR and RT-FSGA.

The remainder of the paper is organized as follows. Section 2 describes the MSVA algorithm and our proposed RT-MSVA implementation, respectively. Section 3 describes the hardware implementation of our newly proposed algorithm, providing details about the architecture of the complete system and the main processing modules. In Sect. 4, we first describe the synthetic and real hyperspectral images used in the experiments and then discuss the obtained processing results and analyses. The computational performance is particularly evaluated in this section. Section 5 concludes the paper with some remarks and hints at plausible future research lines.

2 Real-time maximum simplex volume algorithm (RT-MSVA)

2.1 Volume calculation in N-FINDR and SGA

N-FINDR is one of the most commonly used endmember extraction algorithms. The algorithm uses the volume of the simplex containing the data as a criterion to identify endmembers. The procedure begins with randomly selecting a set of pixels as the initial endmembers. Then, every initial endmember is replaced by other pixels in the image until the maximum volume is found. The randomly chosen initial endmembers may lead to inconsistent results, and the iterative volume calculation adds a heavy burden in terms of computation. SGA intends to address these issues by adding one vertex at a time to the target simplex, until reaching a desired number of endmembers. This reduces the computational complexity tremendously. The initial endmembers are fixed to the two pixels with the largest distance. Although the two algorithms have different iterative procedures, they use the same way to calculate the simplex volume. In other words, they consider that the volume defined by the set of endmembers $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$ can be calculated as follows:

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_p \end{bmatrix}, \tag{2}$$

$$V(\mathbf{E}) = \frac{1}{(p-1)!} abs(|\mathbf{E}|), \tag{3}$$

where the determinant of matrix $\mathbf{E}$ in (2) is calculated using expression (3). This brings the limitation that $\mathbf{E}$ should be a square matrix. Namely, the number of spectral bands of endmembers should be $p - 1$. Generally, the number of endmembers p is much smaller than the number of spectral bands, so a dimensionality reduction step is necessary prior to execution of both N-FINDR and SGA. This not only increases the processing time and computing resources used, but may also remove some potentially relevant information contained in the original data. To overcome these drawbacks, a new volume formula is introduced in the calculation of simplex volume.

2.2 MSVA algorithm

As mentioned in the previous subsection, a square matrix should be created before the determinant-based calculation of the volume. In this section, instead of reducing the dimensionality of the image data, we introduce a new formula that takes advantage of the measure matrix formed by multiplying a non-square matrix by its transpose [14]. Let us suppose that endmembers $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$ compose the vertices of a simplex. If we let $\mathbf{v}_1 = \mathbf{e}_2 - \mathbf{e}_1$, $\mathbf{v}_2 = \mathbf{e}_3 - \mathbf{e}_1, \ldots,$ $\mathbf{v}_{p-1} = \mathbf{e}_p - \mathbf{e}_1$, then $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{p-1}\}$ can be seen as a group of vector bases in the spectral feature space, and the simplex formed by $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$ will be a part of the $p - 1$-dimensional parallel polyhedron expanded by this vector base set. The volume of the polyhedron is closely related to the determinant of the measure matrix:

- When $p - 1 = 1$, the polyhedron is just the line segment defined by vector $\mathbf{v}_1$. The length of the line segment is defined as the norm of $\mathbf{v}_1$, i.e.,

$$V_{\mathrm{p}}(\mathbf{v}_1) = \|\mathbf{v}_1\| = \sqrt{\mathbf{v}_1^{\mathrm{T}}\mathbf{v}_1}, \tag{4}$$

- When $p - 1 = 2$, the proposed polyhedron is a parallelogram expanded by $\mathbf{v}_1$ and $\mathbf{v}_2$. The area is:

$$V_{\mathrm{p}}(\mathbf{v}_1,\mathbf{v}_2) = \|\mathbf{v}_1\| \times \|\mathbf{v}_2\| \times \sin\angle\mathbf{v}_1,\mathbf{v}_2 = \sqrt{\det\left(\begin{pmatrix}\mathbf{v}_1^T\\ \mathbf{v}_2^T\end{pmatrix}(\mathbf{v}_1 \quad \mathbf{v}_2)\right)}, \tag{5}$$

which equals the determinant of the measure matrix.

- For $p-1>2$, it can be proved that the volume of the proposed $p-1$-dimensional parallel polyhedron always equals the square root of the determinant of the measure matrix.

The relation of the volume of simplex (V_{s}) and the volume of the parallel polyhedron (V_{p}) is as follows:

$$V_{\mathrm{s}}(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_{p-1}) = \frac{1}{(p-1)!}V_{\mathrm{p}}(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_{p-1}). \tag{6}$$

Hence, the volume of a simplex whose p vertices are $\mathbf{e}_1,\mathbf{e}_2,\ldots,\mathbf{e}_p$ can be calculated as follows:

$$\mathbf{A} = \mathbf{W}^{\mathbf{T}}\mathbf{W}, \tag{7}$$

$$V(\mathbf{e}_1,\mathbf{e}_2,\ldots,\mathbf{e}_p) = \frac{1}{(p-1)!}\sqrt{|\det(\mathbf{A})|}, \tag{8}$$

where $\mathbf{W} = [\mathbf{e}_2-\mathbf{e}_1,\mathbf{e}_3-\mathbf{e}_1,\ldots,\mathbf{e}_p-\mathbf{e}_1]$ and $\mathbf{A}$ is the measure matrix. Due to the fact that $\mathbf{A}$ is a square matrix in any case, no dimensionality reduction is needed before endmember extraction for a hyperspectral data set with $L(>p)$ bands. This new method of endmember extraction is generally called MSVA [15].

2.3 Real-time MSVA algorithm (RT-MSVA)

Before describing our real-time implementation, it is important to remind that MSVA does not need any dimensionality reduction and that it can effectively exploit all the spectral bands in the original hyperspectral image.

This algorithm has been implemented on GPUs [31] to accelerate the computation. However, calculating the determinant of each matrix still involves significant computational complexity, and this operation is not suitable for real-time applications on FPGAs. To simplify determinant calculation in MSVA, a method based on partitioned determinant operation is proposed in [15]. By using elementary transformation, the measure matrix $\mathbf{A}$ can be partitioned to two diagonal parts. In this way, the determinant of $\mathbf{A}$ can be calculated as the multiplication of a constant partitioned determinant and a changing value depending on the input pixel. This method reduces the computational complexity and enhances the stability as well. However, the inverse matrix of $\mathbf{A}$ must be calculated at the beginning of each outer loop, and this is also a problem for being implemented on FPGAs. To address this issue, we use Woodbury's identity [32] as follows:

$$\left(\mathbf{A}+\mathbf{u}\mathbf{v}^{\mathbf{T}}\right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}}{1+\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u}}, \tag{9}$$

$$\det\left(\mathbf{A}+\mathbf{u}\mathbf{v}^{\mathbf{T}}\right) = \left(1+\mathbf{v}^{\mathbf{T}}\mathbf{A}^{-1}\mathbf{u}\right)\det(\mathbf{A}), \tag{10}$$

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

Let $\mathbf{W}=[\mathbf{e}_2-\mathbf{e}_1,\mathbf{e}_3-\mathbf{e}_1,\ldots,\mathbf{e}_j-\mathbf{e}_1]$, and $\mathbf{A} = \begin{pmatrix}\mathbf{W}^{\mathbf{T}}\mathbf{W} & \mathbf{o}\\ \mathbf{o} & 1\end{pmatrix}$ 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. This means that the inverse matrix and the determinant can be updated by using Woodbury's identity twice in each iteration. The vector definitions are given by:

$$\mathbf{v}_1 = \mathbf{u}_2 = [0,\ldots,0,1]^{\mathbf{T}} \tag{11}$$

$$\mathbf{u}_1 = \left[(\mathbf{e}_2-\mathbf{e}_1)^{\mathbf{T}}(\mathbf{r}-\mathbf{e}_1),\ldots,(\mathbf{r}-\mathbf{e}_1)^{T}(\mathbf{r}-\mathbf{e}_1)\right]^{T} \tag{12}$$

$$\mathbf{v}_2 = \left[(\mathbf{r}-\mathbf{e}_1)^{\mathbf{T}}(\mathbf{e}_2-\mathbf{e}_1),\ldots,(\mathbf{r}-\mathbf{e}_1)^{\mathbf{T}}(\mathbf{e}_j-\mathbf{e}_1),-1\right]^{\mathbf{T}} \tag{13}$$

and the final matrix is given by the following term:

$$\mathbf{A}+\mathbf{u}_1\mathbf{v}_1^{\mathbf{T}}+\mathbf{u}_2\mathbf{v}_2^{\mathbf{T}} = [\mathbf{e}_2-\mathbf{e}_1,\ldots,\mathbf{e}_j-\mathbf{e}_1,\mathbf{r}-\mathbf{e}_1]^{\mathbf{T}}[\mathbf{e}_2-\mathbf{e}_1,\ldots,\mathbf{e}_j-\mathbf{e}_1,\mathbf{r}-\mathbf{e}_1]. \tag{14}$$

It should also be noted that all the determinants in the inner loop are updated from the same initial value *det_old*, which equals to *det_max* in the last outer loop. Instead of updating $V(\mathbf{e}_1,\ldots,\mathbf{e}_{i-1},\mathbf{r}_i)$ from $V(\mathbf{e}_1,\ldots,\mathbf{e}_{i-1},\mathbf{r}_{i-1})$, the formerly 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 in the initialization stage. Selecting a proper set of initial endmembers is an important task for maximum volume algorithms [33]. As reported in [30], the first endmember can be generated by finding the pixel vector with maximum length.

The new method proposed in this paper is called RT-MSVA, which can achieve real-time processing capability as shown in subsequent sections. The RT-MSVA algorithm takes advantage of the simplex growing method in [12]. It first finds an initial endmember and then extracts one endmember in each outer loop. The RT-MSVA can be summarized by the algorithmic description given in Algorithm 1, which is intended to facilitate reproducibility of our results.

Algorithm 1: RT-MSVA

Input: hyperspectral image data $\mathbf{r}$, number of endmembers p

Step1: initialization: calculate the vector length of each pixel r, the pixel that provides the maximum length is the first endmember $\mathbf{e}_1$. Let $i = 0$;

Step2: let $i = i + 1$, and start the inner loop;

Step3: For each pixel $\mathbf{r}$, calculate the volume $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r})$ defined by (7) and (8);

Step4: A variable *det_max* is used to store the largest determinant within this inner loop. Compare $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r})$ with *det_max*, if $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r}) > det\_max$, let $det\_max = V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r})$. The pixel that provides the largest $V(\mathbf{e}_1, ..., \mathbf{e}_{i-1}, \mathbf{r})$ is extracted as endmember $\mathbf{e}_i$;

Step5: if $i \le p$, then go to step 2. Otherwise, the subset $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p\}$ is the set of desired p endmembers;

Output: endmembers $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p\}$.

3 FPGA implementation of RT-MSVA

The hardware architecture of the complete hardware system is shown in Fig. 1. The volume of hyperspectral images is quite large, so a DDR3 synchronous dynamic random access memory (SDRAM) outside the FPGA chip is used to store the whole hyperspectral image. A memory interface generator (MIG) is used to control the SDRAM and register the input data with the help of a first-in-first-out (FIFO) queue. The FIFO has been set to contain two complete rows of the hyperspectral image which builds a ping-pong buffer. Every time a row has been processed (i.e., the FIFO is half empty), and a new data row is brought in. This ensures that the system will not be idle waiting for data and ensures a complete processing workflow. It should be noted that each pixel in the hyperspectral images will be calculated p times throughout the whole process, so the FIFO needs to read the image data p times as well.

The hardware architecture of RT-MSVA can be simply described by three units: the *control unit*, the *determinant calculation module*, and the *compare and update module*. The control unit is used to coordinate all the other units,

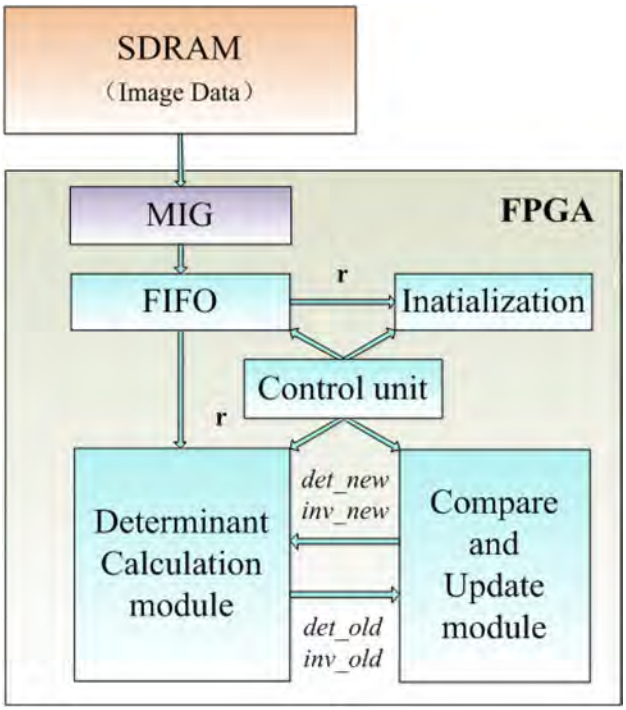

Fig. 1 Hardware system adopted for the implementation of RT-MSVA on FPGAs

namely to determine when the former unit transmits its data to the next unit, and when to get a new pixel from the SDRAM. The determinant calculation module plays an important role in the system, as it represents most of the computation. The compare and update module is used to decide whether the outcome of every loop is effective, and to update the parameters that other units need. A pixel $\mathbf{r}$ is read by the determinant calculation module through the MIG and obtains the determinant *det_new* and the corresponding inverse matrix *inv_new*. In the compare and update module, the newly generated determinant *det_new* is compared with 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. Otherwise, the *det_new* for the current pixel is abandoned and we start the calculation of the next pixel.

Figure 2 describes the data flow of the determinant calculation module. The inputs of this module are the new pixel $\mathbf{r}$, the extracted endmembers, and the intermediate values $\det(\mathbf{A})$ and $\mathbf{A}^{-1}$ (namely *det_old* and *inv_old*). Woodbury's identity should be used twice in each iteration so that the inputs of the second round are effectively the determinant and inverse matrix calculated at the first round. These intermediate values are named as *det_temp* and *inv_temp*. We use the IP core generator to generate the multipliers and dividers. We can choose between using lookup tables (LUTs) or digital signal processors (DSPs) to construct these components in the graphical interface. With

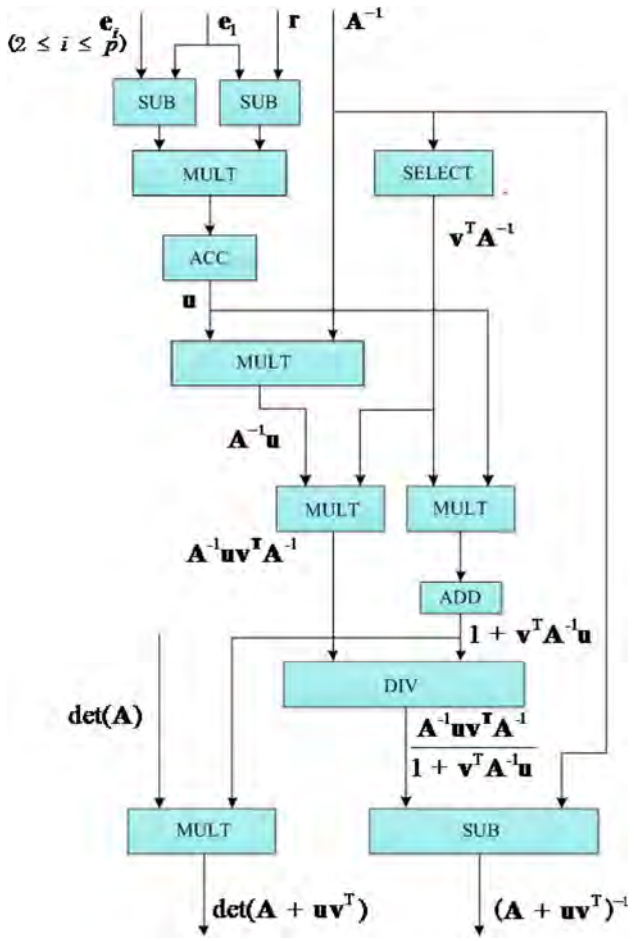

Fig. 2 Data flow of the determinant calculation module

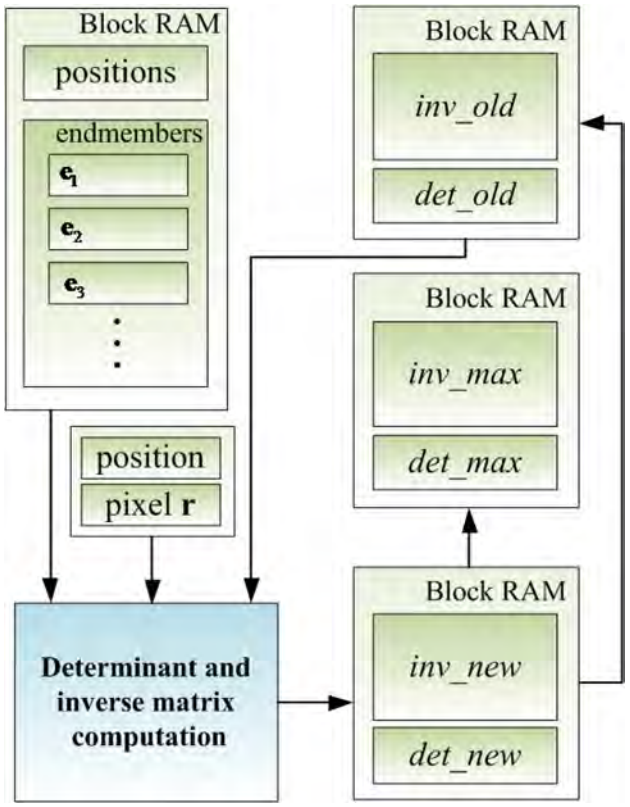

Fig. 3 Storage architecture used for results and intermediate values

the ultimate goal of making full use of the calculation resources on the chip, some components have been reused two or more times within this module. It should be noted that because only the last element in vector $\mathbf{v}$ is 1 and the rest are 0, so the calculation of $\mathbf{v}^T$ multiplied by $\mathbf{A}^{-1}$ can be simplified as selecting the last line of matrix $\mathbf{A}^{-1}$. Figure 3 shows the architecture used to store results and values. As shown in Fig. 3, the new pixel $\mathbf{r}$ is read from the SDRAM and stored in a FIFO, while the other inputs and outputs are all stored in block RAMs. The final output is the set of endmembers which are also stored in a block RAM.

At this point, we provide a step-by-step description of how the proposed algorithm is implemented on our considered hardware architecture.

- First of all, the FIFO brings in two rows of data. Then, it sends them to the initialization module (one by one) to calculate the length of the vector. The pixel vector that provides the maximum length and its position are registered, and this pixel is labeled as the first endmember: $\mathbf{e}_1$. When calculating the last row of data, the first row is brought into the FIFO again for the next step.
- In order to get the second endmember, the control unit orders the FIFO to send data to the initialization module so that the distance between every pixel and the first endmember $\mathbf{e}_1$ can be calculated. The pixel with the largest distance is labeled as the second endmember: $\mathbf{e}_2$. The line segment between $\mathbf{e}_1$ and $\mathbf{e}_2$ can be regarded as a two-dimensional simplex, and the distance corresponds to the volume of the simplex. This distance is stored in variable *det_old* for the next step, and its reciprocal is stored in variable *inv_old*.
- At this point, the initialization stage has finished and the system now finds the ith endmembers ($i = 3, 4, \ldots, p$). The FIFO will send the necessary data to the determinant calculation module. The main part of this module can be reused twice for each pixel. The determinant and inverse matrix obtained in the first round are stored in the block RAM and taken as the inputs to the second round. The final output corresponds to the volume of the simplex formed by pixel $\mathbf{r}$ and the extracted endmembers. In the compare and update module, the absolute value of the determinant is compared with the previously stored *det_max* value. If the first one is greater, then the variables *det_max* and *inv_max* will be updated, and the pixel and its corresponding position are also registered.

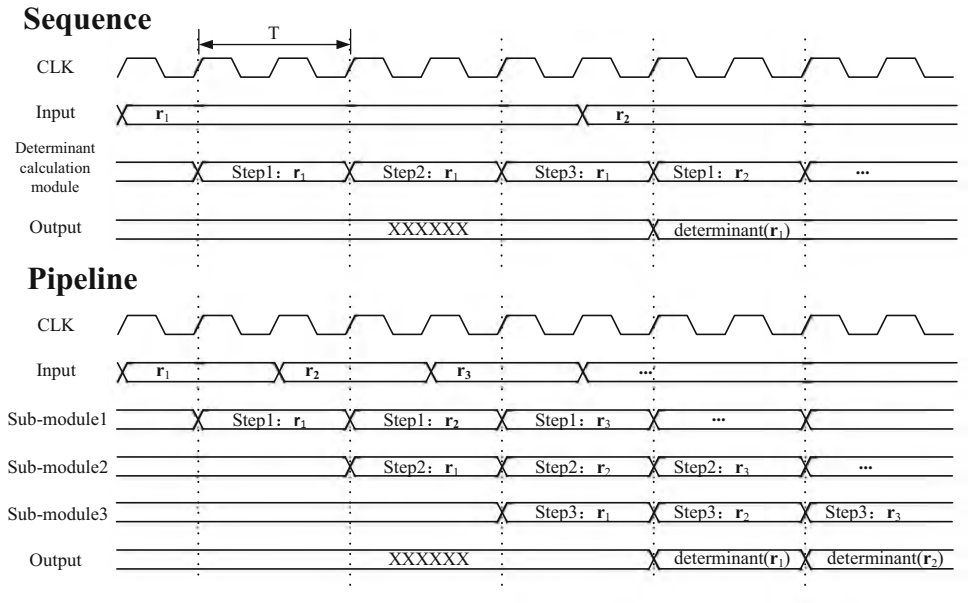

Fig. 4 A schematic view of the pipeline implementation adopted for the determinant calculation module

- When all the pixels in the image have been processed, the registered pixel is extracted as endmember $\mathbf{e}_1$. The new endmember and its position are written in the block RAM. If $i < p$, we let $i = i + 1$. At the same time, the input values *det_old* and *inv_old* are updated by *det_max* and *inv_max* values. If $i = p$, this means that all the endmembers have been extracted, and the process is over. The endmembers and their corresponding coordinates are sent out via the peripheral component interconnect express bus (PCIe).

To further optimize this system, a pipeline implementation has been adopted for the determinant calculation module. Figure 4 shows a schematic view of this approach. Let us assume that the determinant calculation processing chain can be divided into three steps and that the processing time of each step T. For a normal processing mode (in which these steps are carried out one by one in sequence), the determinant calculation module has an output interval of 3*T, but for the pipeline mode (in which these steps are executed simultaneously), the interval can be reduced to T. A FIFO is used to register the output of each step. In brief, pipeline technology can significantly reduce the idle time, thereby gaining a remarkable speedup.

An important issue we must pay attention to is the arithmetical precision. FPGAs can provide both floating-point operations and fixed-point operations. Data in floating-point format can provide more details, but it uses many more computing resources than fixed-point operations. These aspects are crucial for real-time processing. The fixed-point format is simpler and needs less logic resources in terms of both storing the data and performing calculations. This implies that operations in fixed-point format require less processing time. On the other hand, FPGAs and other chips are all discrete systems, which means that the data stream in these chips is given by discrete values. In this implementation, we have used fixed-point format under the assumption that the details that are lost with regard to using floating-point format are negligible in our context.

Figure 5 illustrates the adopted data structure in fixed-point format. To reduce the computational resources and processing time, the data size in this module should be as small as possible under the premise of ensuring accuracy. The input data are given by reflectance units smaller than 1, so we use a 16-bit binary format to represent the data. The highest bit is the sign, and the next bit expresses the integer part, while the lower 14 bits express the fractional part. By multiplying $\mathbf{W}$ by its transpose, every element in $\mathbf{W}^{\mathbf{T}}\mathbf{W}$ and its determinant can be extremely small, and the elements in its inverse matrix may be very large at the same time. As a result, we need to find a compromise for the size of the data. A 32-bit format with 16 bits as fractional part is therefore used to represent elements in the inverse matrix, while a 32-bit format with 24 bits as fraction part is used for the determinant.

4 Experimental results and analysis

The hardware architecture has been implemented on a Xilinx Kintex-7 FPGA KC705 Evaluation Kit (see Fig. 6). The XC7K325T of Kintex-7 series has 50,950 slices, 326,080 logic cells and 407,600 CLB flip-flops available, and also a total block RAM of 16,020 kb. It can also offer

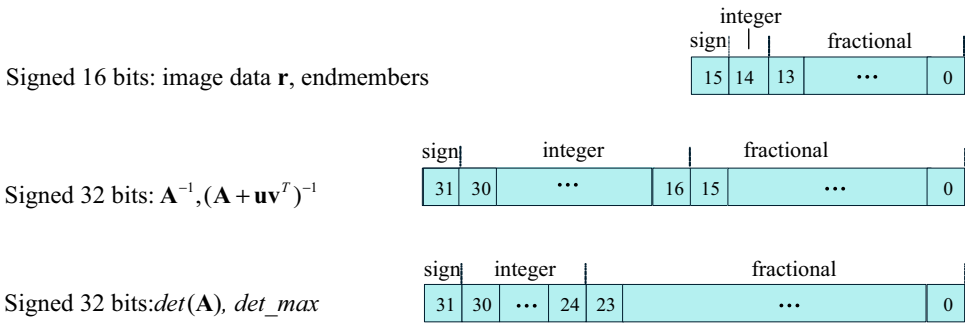

Fig. 5 Data structures used in our FPGA implementation of RT-MSVA

Fig. 6 Xilinx KC705 Evaluation Kit with a XC7K325T FPGA

up to 1866 Mb/s DDR3 data rate, and this makes the memory read/write bandwidth no longer the bottleneck in speeding up. The Kintex-7 family of FPGAs provides an ideal balance in terms of integration, price, performance, and power consumption and can double the price–performance ratio with half the power consumption as compared with the Xilinx Vertex-6 family. The architecture is implemented using the VHDL language. We use ISE 14.2 as the developing environment and Modelsim SE 10.1a to perform simulation. The comparative experiments are conducted on MATLAB in an Intel Core i3 CPU, with a working frequency of 3.5 GHz. Both synthetic and real hyperspectral images will be used to assess endmember extraction accuracy. The real data are also used to analyze the computational performance and the resource utilization of our implementation.

The remainder of this section is organized as follows. In Sect. 4.1, we describe the hyperspectral images that will be used in the experiments. In Sect. 4.2, we discuss the obtained results and provide an evaluation of endmember extraction accuracy in comparison with the MATLAB version. Section 4.3 provides an analysis of the computational performance of our hardware implementation.

4.1 Hyperspectral image data sets

Two sets of hyperspectral images have been used in our experiments. One is a synthetic image, and the other is a real hyperspectral data set. Both data sets have ground truth that can be used to evaluate the endmember extraction accuracy.

4.1.1 Synthetic images generated using fractals

The synthetic images used in this experiment are given by a set of five 100 × 100-pixel hyperspectral scenes generated using fractals. These synthetic hyperspectral scenes have been created as follows. First, random spatial patterns are generated using fractals. Then, these images are divided into a set of clusters, using the k-means algorithm. Finally, a set of spectral signatures are randomly selected from the USGS spectral library and assigned to each cluster. The purest pixels are arranged in the center of each region, while pixels close to the edge of regions are heavily mixed [34]. The number of endmembers is fixed to $p = 9$. Figure 7 shows a false color image of the five synthetic scenes in the experiments. Figure 8 shows the nine spectral signatures present in the first synthetic scene, called Fractal 1

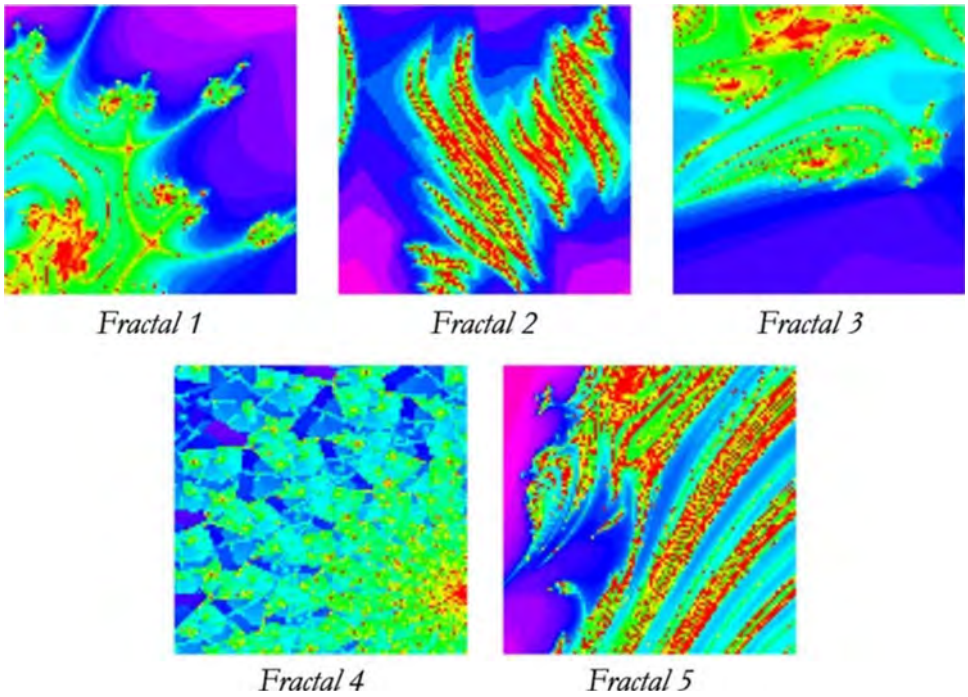

Fig. 7 False color composition of the five synthetic images used in experiments

(namely, alunite, dumortierite, halloysite, kaolinite CM9, kaolinite KGa1, muscovite, nontronite, pyrophilite, and sphene).

4.1.2 Real hyperspectral image

The real hyperspectral image used in our experiments was collected by the AVIRIS sensor [35] over the Cuprite mining district, Nevada, in 1997. The data are available in reflectance units after atmospheric correction and have 224 spectral bands in the range from 400 to 2500 nm. A subset of 350 × 350 pixels is used in this experiment. According to [30], 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. This site has been well studied mineralogically and has five representative pure minerals in this region: *alunite*, *buddingtonite*, *calcite*, *kaolinite*, and *muscovite*. The spectral signatures of these minerals are available in the USGS spectral library and will be used in the following section to assess endmember extraction accuracy. The virtual dimensionality (VD) is used in this work to estimate the number of endmembers in the AVIRIS Cuprite image, which is calculated to be $p = 22$ as reported in [36, 37].

4.2 Endmember extraction accuracy analysis

In this subsection, we evaluate the endmember extraction accuracy of our proposed hardware implementation of RT-MSVA using the synthetic and real data sets described above. Two metrics have been widely used to quantitatively evaluate endmember extraction accuracy. One is comparing the endmembers extracted by the proposed algorithms with the reference spectral signatures using the spectral angle (SA) [38]. Another one is reconstructing the hyperspectral image by using the extracted endmembers and comparing it with the original image using the root-mean-square error (RMSE). Each endmember or reference spectral signature can be seen as a vector, so the SA is calculated as the vector angle between the most similar endmember extracted and the pure spectral signature

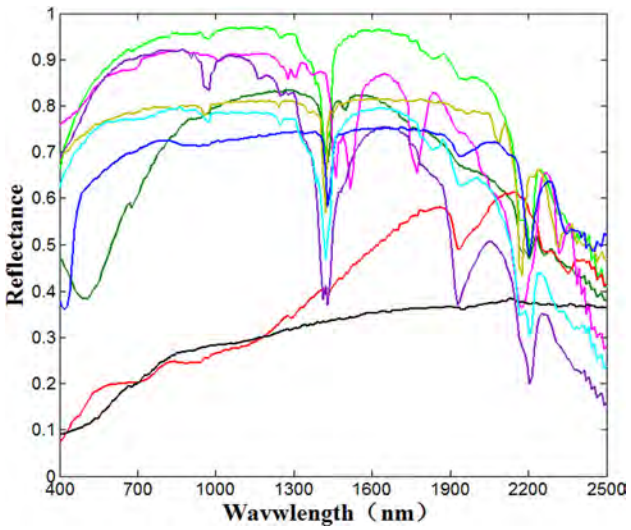

Fig. 8 The nine USGS library signatures used to create the synthetic scene called "Fractal 1"

contained in the library. A smaller SA score indicates higher similarity between the two vectors, which means that the endmember spectrum is closer to the reference signature. In this work, the fully constrained least squares (FCLS) [39] method is used to get the corresponding abundances for each endmember. As shown in Eq. (1), a hyperspectral image can be rebuilt by multiplying endmember matrix **E** and the abundance matrix **A**. The reconstructed image has some differences with the original image, and the RMSE accounts for the error between these two images. Low RMSE means high similarity between the reconstructed and the original image, which further indicates better endmember extraction accuracy. Before describing our experiments, we emphasize that we have included comparisons to other algorithms such as N-FINDR, SGA, and MSVA (implemented in MATLAB).

Table 1 shows the average SA scores between the endmembers extracted by the aforementioned algorithms and the USGS reference spectral signatures across all the five synthetic scenes. The RMSE obtained in the reconstruction using the extracted endmembers is also reported. The best score for each metric is bolded in the table. Nine endmembers are extracted and each of them corresponds to a USGS library signature known a priori. The SA score of each algorithm in Table 1 is the average SA of all nine endmembers, which is given in degrees. For the N-FINDR algorithm, the initial set of endmembers is randomly selected and this makes the result inconsistent. To have a fair comparison, the experiments with N-FINDR have been conducted five times with the same image data and parameters, and we take the average value as the final outcome. The hardware implementation of RT-MSVA gives exactly the same result as the software version for the considered synthetic scenes.

Figure 9a, b, respectively, shows the endmembers extracted from the AVIRIS Cuprite scene by RT-MSVA using MATLAB and our FPGA-based implementation. The endmembers are marked in the same order as they were extracted. As shown in Fig. 9, for the first 13 endmembers the FPGA design gets exactly the same result as MATLAB does, and then, there are some differences in the extracted endmembers. These differences result from the accumulated calculation error in the hardware implementation. This is because the original hyperspectral image data are stored in double float data type, with 64 bits to represent each value. However, as shown in Fig. 2, the image data are stored using 16 bits in our data structure, which means that it is unavoidable that a part of the fractions may be cut off. Besides, all the intermediate variables face the same circumstance. These differences are very small and can be negligible in many circumstances, but after a number of iterations they become more and more relevant, to the extent that this influences the final result and we obtain a different set of endmembers. As mentioned in Sect. 3, the adopted data structure has been designed with the ultimate goal of optimizing speed and resource utilization. If we increase the number of bits used to represent the values during the processing, the calculation error can be constrained to small values (at the cost of using more computing resources in the FPGA). However, we have experimentally found that the endmembers obtained using the obtained structure are equally relevant, as indicated in our assessment below.

Table 1 Average spectral angle scores (in degrees) between the endmembers extracted by several algorithms and the USGS reference spectral signatures for the synthetic scenes. The RMSE obtained in the reconstruction using the extracted endmembers is also reported

Algorithms	Spectral angle (degrees)	RMSE
N-FINDR	0.362	$\mathbf{4.11 \times 10^{-4}}$
SGA	0.459	6.34×10^{-4}
MSVA	**0.357**	4.63×10^{-4}
RT-MSVA	**0.357**	4.63×10^{-4}

The SA and RMSE scores obtained for the AVIRIS Cuprite image are shown in Table 2. The number of endmembers extracted is much larger than the number of USGS reference signatures, so we choose the smallest SA for each mineral as the final score. The best score for each metric is bolded in the table. As shown in Table 2, N-FINDR, MSVA, and RT-MSVA can obtain much lower RMSEs than SGA, and among them RT-MSVA results in the lowest RMSE score. On the other hand, RT-MSVA performs well in terms of SA, obtaining, for instance, the best matching score for the buddingtonite, calcite and kaolinite minerals. N-FINDR exhibits good accuracy because it traverses all the pixel combinations, but this takes a very long time. MSVA uses a simplex growing method and exhibits higher computational efficiency than N-FINDR. In general, MSVA outperforms SGA and N-FINDR if we take both accuracy and computational efficiency into consideration. RT-MSVA is the FPGA-implemented version of MSVA. As mentioned before, there are differences in the obtained endmembers resulting from accumulated computing errors due to the adopted data structure. But the errors are far from fatal as the obtained scores are not far from the optimal ones. For instance, in our experiment with AVIRIS Cuprite data, the FPGA-based implementation of RT-MSVA gets a RMSE that is very close to the one reported for MSVA

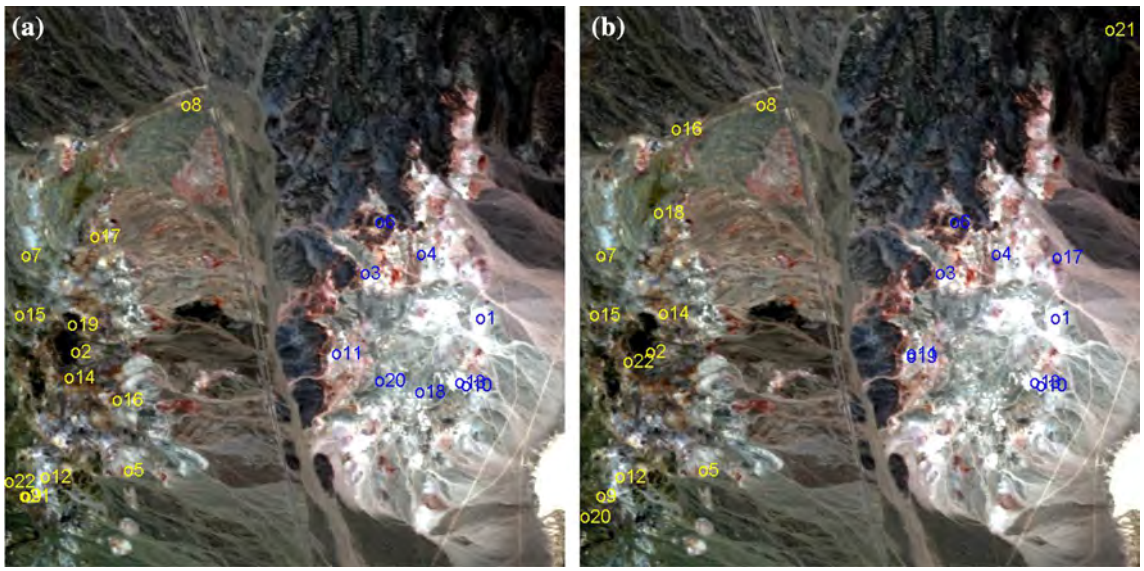

Fig. 9 Endmembers extracted by the proposed RT-MSVA algorithm from the AVIRIS Cuprite scene. **a** MATLAB version. **b** FPGA version

Table 2 Average spectral angle scores (in degrees) between the endmembers extracted by several algorithms and the USGS reference spectral signatures for the real AVIRIS Cuprite scene. The RMSE obtained in the reconstruction using the extracted endmembers is also reported

Algorithms	Alunite	Buddingtonite	Calcite	Kaolinite	Muscovite	RMSE
N-FINDR	**2.106**	5.781	4.390	2.343	2.843	0.0049
SGA	2.501	6.328	5.294	2.634	**2.298**	0.0065
MSVA	2.917	**5.610**	**4.037**	**1.766**	2.687	0.0047
RT-MSVA	4.113	**5.610**	**4.037**	**1.766**	2.687	**0.0046**

Table 3 Summary of resource utilization of the FPGA implementation of RT-MSVA for the AVIRIS Cuprite data

Logic utilization	Used	Available	Utilization (%)
Number of slice registers	37,140	407,600	9
Number of slice LUTs	69,286	203,800	34
Number of fully used LUT-FF pairs	17,864	76,140	23
Number of bonded IOBs	33	500	6
Number of block RAM/FIFO	148	445	33
Number of BUFG/BUFGCTRLs	1	32	3
Number of DSP48E1s	567	840	67
Maximum frequency	48.260 MHz		

and even gets better performance in terms of SA scores. In conclusion, although our hardware implementation of RT-MSVA has some accumulated errors, it still keeps better balance between accuracy and computational performance than N-FINDR and SGA.

4.3 Computational performance evaluation

In this subsection, we will analyze the computational performance of the proposed FPGA-based implementation. The resource utilization is summarized in Table 3. Because we use block RAMs to store the data, the cost of slice registers and slice LUTs is not very high. The main limitation of our implementation is the number of DSPs (of type DSP48E1) that are used to implement the computing units such as multipliers and dividers. It should be noted that the resource utilization is not constant when the data set to be processed changes. In other words, for different numbers of spectral bands, different numbers of operators should be arranged.

Table 4 shows the computing times reported for RT-MSVA and RT-FSGA when processing the AVIRIS Cuprite data by our FPGA-based implementation and the equivalent software version in MATLAB. The

Table 4 Processing time (seconds) measured for the proposed RT-MSVA with the AVIRIS Cuprite image (MATLAB and FPGA implementation)

	MATLAB (s)	FPGA	
		Processing time (s)	Clock periods
RT-FSGA	46.58	0.16	3.11×10^7
RT-MSVA	163.38	0.84	4.06×10^7

experimental results of RT-FSGA implemented on an FPGA are reproduced from [27]. For RT-MSVA, MATLAB used 163.38 s to obtain the endmembers, and our FPGA implementation used $0.4.06 \times 10^7$ clock periods. The clock frequency is 48.260 MHz, and the processing time is just 0.84 s. Compared to 0.16 s of RT-FSGA, the implementation of RT-MSVA is a bit slower. However, as discussed before, RT-MSVA exhibits better accuracy than RT-FSGA in terms of endmember extraction. Moreover, RT-MSVA does not need a dimensionality reduction step, 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 17 MB/s. The data size of the real hyperspectral image used in this paper is about 55 MB, so the new generation AVIRIS sensor requires at least 3.23 s to acquire an image with the same size. Our FPGA implementation can complete the endmember extraction in 0.84 s, 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 may still be further optimized in future developments.

5 Conclusions and future lines

Hyperspectral imaging applications often require fast processing techniques able to perform in (near) real time. However, high spectral resolution leads to high computational complexity. As a result, there is an urgent need to develop real or near real-time implementations for hyperspectral image processing. In this paper, we have presented an FPGA-based implementation of a maximum simplex volume algorithm (RT-MSVA) for endmember extraction from hyperspectral images. This method does not require a previous dimensionality reduction, and it can take full advantage of the information contained in the original data cube. The simplex growing method and a fast matrix determinant computation greatly help to decrease the computational complexity. Our experimental results indicate that the proposed hardware implementation of RT-MSVA exhibits better accuracy than other FPGA implementations such as RT-FSGA (based on simplex growing). Our proposed system can fully achieve real-time processing capabilities, which is remarkable in FPGA systems as this paves the way for spaceborne hyperspectral data processing with reconfigurability, low power consumption, and radiation tolerance. Additional experiments with other hyperspectral data sets will be conducted in future developments to further discuss the influence of a varying number of spectral bands in the input data. We will also further explore the balance between accuracy, speed, and computing resources for the proposed implementations. Also, we consider proving the applicable ability of FPGA-based real-time maximum simplex volume algorithm by using images of larger size, i.e., of size that is closer or equal to the real hyperspectral images. Moreover, we want to integrate abundance estimation module when we carry on this research. Another future research line will be to reduce the number of pixels to be calculated by preprocessing [40], which may also lead to a substantial decrease in processing time.

Acknowledgements This research was supported by the National Natural Science Foundation of China under Grant Nos. 41325004, 41571349, and 91638201.

References

1. Goetz, A.F.H., Vane, G., Solomon, J.E., Rock, B.N.: Imaging spectrometry for earth remote sensing. Science **228**(4704), 1147–1153 (1985)
2. Jia, S., Xie, Y., Tang, G., Zhu, J.: Spatial-spectral-combined sparse representation-based classification for hyperspectral imagery. Soft. Comput. **45**, 101–110 (2014)
3. Qian, Y., Yao, F., Jia, S.: Band selection for hyperspectral imagery using affinity propagation. IET Comput. Vis. **3**(4), 213–222 (2009)
4. Dias, J.M.B., Plaza, A., Dobigeon, N., Parente, M., et al.: Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **5**(2), 354–379 (2012)
5. Keshava, N., Mustard, J.F.: Spectral unmixing. IEEE Signal Process. Mag. **19**, 44–57 (2002)
6. Chang, C.-I.: Target abundance-constrained subpixel detection: Partially Constrained Least-Squares Methods. In: Hyperspectral Imaging. Springer, US, pp. 39–50 (2003)
7. Plaza, A., Martínez, P., Pérez, R., Plaza, J.: A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. IEEE Trans. Geosci. Remote Sens. **42**(3), 650–663 (2004)
8. Boardman, J.W.: Geometric mixture analysis of imaging spectrometry data. In: Proceedings of International Geoscience Remote Sensing Symposium, Pasadena, CA, vol. 4, pp. 2369–2371. (1994)
9. Nascimento, J.M.P., Dias, J.M.: Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Trans. Geosci. Remote Sens. **43**(4), 898–910 (2005)

10. Neville, R.A., Staenz, K., Szeredi, T., Lefebvre, J., Hauff, P.: Automatic endmember extraction from hyperspectral data for mineral exploration. In: Proceedings of 4th International Airborne Remote Sensing Conference and Exhibition/21 st Canadian Symposium Remote Sensing, Ottawa, ON, Canada, June, pp. 21–24. (1999)
11. Winter, M.E.: N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data. In: Proceedings of SPIE, vol. 3753, pp. 266–275. (1999)
12. Chang, C.-I., Wu, C., Liu, W., Ouyang, Y.C.: A growing method for simplex-based endmember extraction algorithms. IEEE Trans. Geosci. Remote Sens. **44**(10), 2804–2819 (2006)
13. Geng, X.R.: Target detection and classification for hyperspectral imagery. Ph.D. dissertation, Institute of Remote Sensing Applications Chinese Academy of Science, Beijing, China (2005)
14. Geng, X.R., Zhao, Y.C., Wang, F.X., Gong, P.: A new formula for a simplex and its application to endmember extraction for hyperspectral image analysis. Int. J. Remote Sens. **31**(4), 1027–1035 (2010)
15. Qu, H., Huang, B., Zhang, J., Zhang, Y.: An improved maximum simplex volume algorithm to unmixing hyperspectral data. In: Proceedings of SPIE, vol. 8895, pp. 889507-1–889507-7. (2013)
16. Zhang, B.: Intelligent remote sensing satellite system. J. Remote Sens. **15**(3), 415–422 (2011)
17. Lee, C.A., Gasster, S.D., Plaza, A., Chang, C.-I., Huang, B.: Recent developments in high performance computing for remote sensing: a review. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **4**(3), 508–527 (2011)
18. Plaza, A., Plaza, J., Paz, A., Sánchez, S.: Parallel hyperspectral image and signal processing. IEEE Signal Process. Mag. **28**, 119–126 (2011)
19. Sánchez, S., Paz, A., Martin, G., Plaza, A.: Parallel unmixing of remotely sensed hyperspectral images on commodity graphics processing units. Concur. Comput. Pract. Exp. **23**(13), 1538–1557 (2011)
20. Lysaght, P., Blodget, B., Mason, J., Young, J., Bridgford, B.: Enhanced architectures, design methodologies and CAD tools for dynamic reconfiguration of Xilinx FPGAs. In: Proceedings of International Conference on Field Programmable Logic Applications, pp. 1–6. (2006)
21. Compton, K., Hauck, S.: Reconfigurable computing: a survey of systems and software. ACM Comput. Surv. **34**, 171–210 (2002)
22. Tessier, R., Burleson, W.: Reconfigurable computing for digital signal processing: a survey. J. VLSI Signal Process. Syst. **28**(1), 7–27 (2001)
23. Sánchez, S., Rui, R., Sousa, L., et al.: Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs. J. Real-Time Image Proc. **10**(3), 469–483 (2015)
24. Sánchez, S., Plaza, A.: Fast determination of the number of endmembers for real-time hyperspectral unmixing on GPUs. J. Real-Time Image Proc. **9**(3), 397–405 (2012)
25. Hauck, S.: The roles of FPGAs in reprogrammable systems. Proc. IEEE **86**(4), 615–639 (1998)
26. Plaza, A., Du, Q., Chang, Y.-L., King, R.L.: High performance computing for hyperspectral remote sensing. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **4**(3), 528–544 (2011)
27. Gonzalez, C., Lopez, S., Mozos, D., et al.: A novel FPGA-based architecture for the estimation of the virtual dimensionality in remotely sensed hyperspectral images. J. Real-Time Image Proc. **43**(5), 1–12 (2015)
28. González, C., Mozos, D., Resano, J., Plaza, A.: FPGA implementation of the N-FINDR algorithm for remotely sensed hyperspectral image analysis. IEEE Trans. Geosci. Remote Sens. **50**(2), 374–388 (2012)
29. Wu, C.-C., Chen, H.-M., Chang, C.-I.: Real-time N-finder processing algorithms for hyperspectral imagery. J. Real-Time Image Proc. **7**(2), 105–129 (2012)
30. Chang, C.-I., Xiong, W., Wu, C.C.: Field-programmable gate array design of implementing simplex growing algorithm for hyperspectral endmember extraction. IEEE Trans. Geosci. Remote Sens. **51**(3), 1693–1700 (2013)
31. Qu, H., Zhang, J., Lin, Z., Chen, H., Huang, B.: GPU acceleration of the simplex volume algorithm for hyperspectral endmember extraction. In: Proceedings SPIE, vol. 8539, pp. 85390B-1–85390B-7. (2012)
32. Xiong, W., Wu, C.C., Chang, C.-I., Kapalkis, K., Chen, H.M.: Fast algorithms to implement N-FINDR for hyperspectral endmember extraction. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **4**(3), 545–564 (2011)
33. Plaza, A., Chang, C.-I.: Impact of initialization on design of endmember extraction algorithms. IEEE Trans. Geosci. Remote Sens. **44**(11), 3397–3407 (2006)
34. Martín, G., Plaza, A.: Region-based spatial preprocessing for endmember extraction and spectral unmixing. IEEE Geosci. Remote Sens. Lett. **8**(4), 745–749 (2011)
35. Green, R.O., Eastwood, M.L., Sarture, C.M., Chrien, T.G., Aronsson, M., Chippendale, B.J., Faust, J.A., Pavri, B.E., Chovit, C.J., Solis, M., Olah, M.R., Williams, O.: Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). Remote Sens. Environ. **65**(3), 227–248 (1998)
36. Chang, C.-I., Du, Q.: Estimation of number of spectrally distinct signal sources in hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. **42**(3), 608–619 (2004)
37. Chang, C.-I., Wu, C.C., Lo, C.-S., Chang, M.-L.: Real-time simplex growing algorithms for hyperspectral endmember extraction. IEEE Trans. Geosci. Remote Sens. **40**(4), 1834–1850 (2010)
38. Chang, C.-I.: Hyperspectral Imaging: Techniques for Spectral Detection and Classification. Kluwer, New York (2003)
39. Heinz, D.C., Chang, C.-I.: Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. **39**(3), 529–545 (2001)
40. Zhao, C., Zhao, G., Qi, B., Li, X.: Reduced near border set for endmember extraction. Optik Int. J. Light Electron Opt. **126**(23), 4424–4431 (2015)

Cong Li received his B.S. degree in communication engineering from Nanjing University, Nanjing, China, in 2013. He is currently pursuing the Ph.D. degree in the Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing, China. His research interests include hyperspectral remote sensing, hyperspectral unmixing, hyperspectral image processing, and high-performance computing with FPGAs.

Lianru Gao received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in cartography and geographic information system from Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2007. He is currently a Professor with the Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences. He has published over 90 papers in China and abroad. His research interests include hyperspectral image processing and information extraction. He is a recipient of the recognition of Best Reviewers of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing in 2015.

Antonio Plaza was born in Caceres, Spain, in 1975. He is an Associate Professor (with accreditation for Full Professor) with the Department of Technology of Computers and Communications, University of Extremadura, where he is the Head of the Hyperspectral Computing Laboratory (HyperComp), one of the most productive research groups working on remotely sensed hyperspectral data processing worldwide. His main research interests comprise hyperspectral data processing and parallel computing of remote sensing data. He has been the advisor of 12 Ph.D. dissertations and more than 30 M.Sc. dissertations. He was the Coordinator of the Hyperspectral Imaging Network, a European project with total funding of 2.8 million Euro. He has authored more than 500 publications, including 152 journal papers (more than 100 in IEEE journals), 22 book chapters, and over 240 peer-reviewed conference proceeding papers (94 in IEEE conferences). He has edited a book on High-Performance Computing in Remote Sensing for CRC Press/Taylor and Francis and guest edited nine special issues on hyperspectral remote sensing for different journals. Dr. Plaza is a Fellow of IEEE "for contributions to hyperspectral data processing and parallel computing of Earth observation data." He is a recipient of the recognition of Best Reviewers of the IEEE Geoscience and Remote Sensing Letters (in 2009) and a recipient of the recognition of Best Reviewers of the IEEE Transactions on Geoscience and Remote Sensing (in 2010), a journal for which he served as Associate Editor in 2007–2012. He is also an Associate Editor for IEEE Access and was a member of the Editorial Board of the IEEE Geoscience and Remote Sensing Newsletter (2011–2012) and the IEEE Geoscience and Remote Sensing Magazine (2013). He was also a member of the steering committee of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS). He is a recipient of the 2013 Best Paper Award of the JSTARS journal, and a recipient of the most highly cited paper (2005–2010) in the Journal of Parallel and Distributed Computing. He received best paper awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He is a recipient of the Best Ph.D. Dissertation award at University of Extremadura, a recognition also received by six of his Ph.D. students. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) in 2011–2012 and is currently serving as President of the Spanish Chapter of IEEE GRSS (since November 2012). He has served as a proposal evaluator for the European Commission, the National Science Foundation, the European Space Agency, the Belgium Science Policy, the Israel Science Foundation, and the Spanish Ministry of Science and Innovation. He has reviewed more than 500 manuscripts for over 50 different journals. He is currently serving as the Editor-in-Chief of the IEEE Transactions on Geoscience and Remote Sensing journal. Additional information: http://www.umbc.edu/rssipl/people/aplaza.

Bing Zhang received the B.S. degree in geography from Peking University, Beijing, China, the M.S. and Ph.D. degrees in remote sensing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences (CAS), Beijing, China. Currently, he is a Full Professor and the Deputy Director of the Institute of Remote Sensing and Digital Earth (RADI), Chinese Academy of Sciences (CAS), where he has been leading key scientific projects in the area of hyperspectral remote sensing for more than 20 years. His research interests include the development of Mathematical and Physical models and image processing software for the analysis of hyperspectral remote sensing data in many different areas, such as geology, hydrology, ecology, and botany. He has developed five software systems in the image processing and applications. His creative achievements were rewarded eight important prizes from Chinese government and special government allowances of the Chinese State Council. He was awarded the National Science Foundation for Distinguished Young Scholars of China in 2013. He was also awarded as one of the top four scientists among more than 100 professors in RADI in 2015 by Chinese Academy of Sciences. He has authored more than 300 publications, including 171 journal papers. He has edited six books/contributed book chapters on hyperspectral image processing and subsequent applications, and those books are serve as the main materials for education and research in hyperspectral remote sensing in China. He has been a professor of University of Chinese Academy of Sciences and gave lessons of hyperspectral remote sensing for more than 10 years. He has been the advisor of 29 Ph.D. dissertations and more than 12 M.S. dissertations. He is currently serving as the Associate Editor for IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (IEEE JSTARS) since 2011, and he is also the guest editor for IEEE JSTARS special issue on "Hyperspectral Remote Sensing: Theory, Methods, and Applications" in 2013, IEEE JSTARS special issue on "Big Data in Remote Sensing" in 2015, and Pattern Recongition Letters (PRL) special issue on "Advances in Pattern Recognition in Remote Sensing" in 2015. He has served as Technical Committee Member of IEEE Workshop on Hyperspectral Image and Signal Processing (IEEE WHISPERS) since 2011, and as the Chair of China National Committee of International Society for Digital Earth since 2012. He is the Student Paper Competition Committee member in IGARSS 2015 and 2016 and Scientific Committee Member of IGRASS in 2014.