Research Article

Fast Endmember Extraction for Massive Hyperspectral Sensor Data on GPUs

Zebin Wu,1,2 Shun Ye,1 Jie Wei,1 Zhihui Wei,1,3 Le Sun,1 and Jianjun Liu1

1 School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
2 Lianyungang Research Institute of NJUST, Lianyungang 222006, China
3 Jiangsu Key Lab of Spectral Imaging and Intelligent Sensing, Nanjing 210094, China

Correspondence should be addressed to Zebin Wu; zebin.wu@gmail.com

Received 14 July 2013; Accepted 12 September 2013

Academic Editor: Zhijie Han

Copyright © 2013 Zebin Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hyperspectral imaging sensor becomes increasingly important in multisensor collaborative observation. The spectral mixture problem seriously influences the efficiency of hyperspectral data exploitation, and endmember extraction is one of the key issues. Due to the high computational cost of algorithm and massive quantity of the hyperspectral sensor data, high-performance computing is extremely demanded for those scenarios requiring real-time response. A method of parallel optimization for the well-known N-FINDR algorithm on graphics processing units (NFINDR-GPU) is proposed to realize fast endmember extraction for massive hyperspectral sensor data in this paper. The implements of the proposed method are described and evaluated using compute unified device architecture (CUDA) based on NVIDIA Quadra 600 and Tesla C2050. Experimental results show the effectiveness of NFINDR-GPU. The parallel algorithm is stable for different image sizes, and the average speedup is over thirty times on Tesla C2050, which satisfies the real-time processing requirements.

1. Introduction

Multisensor image data fusion in remote sensing is a kind of collaborative image processing technology for sensor networks, which utilize the consistency and complementarity of different sensors’ image data to assess accurately [1]. Hyperspectral imaging sensor becomes increasingly important in multisensor collaborative observation. There are tens or hundreds of contiguous bands of high spectral resolution in hyperspectral image, covering the visible, near-infrared, and shortwave infrared spectral bands [2]. It can get the spectral signatures and enable identification of the materials that make up a scanned target, which greatly improve the ability of target recognition and detection in sensor networks.

Spatial resolution of hyperspectral sensor data is often relative low to meters or tens of meters, and several different materials jointly occupy a single pixel. Therefore, most of the pixels of hyperspectral data, which called mixed pixels, contain more than one material (called endmember) [3]. The mixture problem will seriously influence the efficiency of hyperspectral data exploitation. Many researchers focus on the study of hyperspectral unmixing. Endmember extraction is one of the key issues. Many algorithms have been developed to solve this issue, and the N-FINDR [4] is one of the most widely used methods for automatically determining endmembers in hyperspectral data without using a priori information, which has been successfully applied for over ten years.

There are two factors which cause the limitation of N-FINDR’s applications in multisensor collaborative observation. First, it is quite expensive in computational terms due to the high algorithm complexity. Second, the quantity of hyperspectral sensor data is massive due to the extremely high dimensionality of hyperspectral data cube. For example, the airborne visible-infrared imaging spectrometer (AVIRIS) is able to record the visible and near-infrared spectrum (wavelength region 400–2500 nm) of reflected light in an area, 2–12 km wide and several kilometers long, using 224 spectral bands. The resulting multidimensional data cube typically comprises severalGBs per flight [5]. High-performance computing is extremely demanded for those scenarios requiring real-time response, such as military detection, monitoring of chemical contamination, and wildfire tracking.
Several parallel computing technologies, like supercomputers, clusters, distributed computing, multicore CPUs, field-programmable gate arrays (FPGAs), and graphics processing units (GPUs), are used to accelerate hyperspectral data processing algorithms [6, 7]. GPUs are quickly evolving as a standardized architecture in hyperspectral processing due to their compactness, low cost, portability, low weight, and high computational power [8]. Wu et al. [9] presented an improved GPU implementation of the PPI algorithm which provides real-time performance. Plaza et al. [10] developed three new GPU-based implementations of endmember extraction algorithms: the pixel purity index (PPI), a kernel version of the PPI (KPI), and the automatic morphological endmember extraction (AMEE) algorithm, and they provided a GPU-based implementation of the fully constrained linear spectral unmixing algorithm. Barberis et al. [6] proposed a new parallel implementation of the vertex component analysis (VCA) algorithm for spectral unmixing of remotely sensed hyperspectral data on commodity GPUs. Although there are several parallel implements of the N-FINDR algorithm existing in literature [5,11–13], the speedup of them is less than 30 times, which cannot meet the requirements of real-time applications well.

In this paper, we propose a method of parallel optimization for N-FINDR on graphics processing units (NFINDR-GPU) to realize real-time endmember extraction for massive hyperspectral sensor data. The implements of the proposed methods using compute unified device architecture (CUDA) are described and evaluated. The computation time of the parallel implementation on GPUs is compared with the serial implementation on central processing units (CPUs).

2. Endmember Extraction for Hyperspectral Sensor Data

There are two models to unmix the hyperspectral sensor data, named linear and nonlinear. The linear mixture model identifies a collection of spectrally pure constituent spectra (endmembers) and expresses the measured spectrum of each mixed pixel as a linear combination of endmembers weighted by fractional abundances that indicate the proportion of each endmember present in the pixel [3]. It assumes minimal secondary reflections and multiple scattering effects in the data collection procedure, and hence the measured spectra can be expressed as a linear combination of the spectral signatures of materials present in the mixed pixel. The linear mixture model is formulated as follows:

\[ y = Ma + n, \]  

where \( y \) denotes a \( L \)-by-1 spectrum vector of one pixel of the observed hyperspectral data; \( L \) denotes the number of bands, \( M = [m_1, m_2, \ldots, m_q] \) denotes a \( L \)-by-\( q \) mixing matrix with endmembers as columns and is usually of full column rank, \( q \) denotes the number of endmembers; \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_q]^T \) denotes a \( q \)-by-1 vector containing the respective fractional abundances of the endmembers, \( \alpha_k \) is the abundance of the \( k \)th endmember, with \( k = 1, 2, \ldots, q \), and the notation \((\cdot)^T\) stands for vector transposed; \( n \) denotes an additive \( L \)-by-1 noise vector collecting the errors affecting the measurements of the pixel at each spectral band. Endmember extraction of hyperspectral data aims at obtaining a good estimation of the mixing matrix \( M \).

Several methods have been used to perform endmember extraction, including geometrical, statistical, and sparse regression-based approaches [3]. A successful and widely used algorithm in the first category has been the N-FINDR.

N-FINDR algorithm is an iterative optimization procedure that maximizes the volume of a simplex containing all hyperspectral image pixels in feature space and automatically extracts the endmembers in the hyperspectral scene [4]. It relies on the assumption that, when the noise vector \( n \) is negligible, all the spectrum vectors of hyperspectral pixels are contained in a convex set (named simplex) of high-dimensional space, and the endmembers are vertices of the simplex. Thus, the problem of endmember extraction is transformed to solving the vertices of the simplex.

The N-FINDR looks for the set of pixels with the largest possible volume by inflating a simplex inside the data. The mathematical definition of the volume \( V \) of a simplex is formulated as

\[ V(E) = \frac{1}{(n-1)!} \abs{\det(E)}, \]  

where \( E \) is the matrix of endmembers augmented with a row of ones, \( E = [e_1, e_2, \ldots, e_q]; e_i \) is a column vector containing the spectra of endmember \( i \); \( \abs{x} \) is the absolute value of \( x \); \( \det(E) \) denotes the determinant of matrix; \((n - 1)\) is the number of dimensions occupied by the data.

The determinant is only defined in the case where the number of features is \( q - 1 \); \( q \) is the number of desired endmembers. Since \( n \gg p \) typically in hyperspectral data, we adopt principal component analysis (PCA) to reduce the dimensionality of the input hyperspectral data [5] and use the virtual dimensionality (VD) method [14] to estimate the number of endmembers, for the preprocessing of N-FINDR. After the preprocessing, the standard N-FINDR algorithm can be summarized as in Algorithm 1.

3. Fast Endmember Extraction Based on GPUs

It could be noted that there are three factors which lead to high computational overhead of the N-FINDR algorithm. Firstly, the algorithm is an iterative procedure, and every pixel in the data set must be evaluated to refine the estimate of endmembers, looking for the set of pixels that maximizes the volume of the simplex defined by the selected endmembers [5]. Secondly, in step (4), the computation is done for every single element in the input data set, and the replacement step is repeated for all the pixel vectors in the dataset. Thirdly, in step (3), the calculation of the determinants is particularly time consuming.

Aiming at solving these problems, the endmember extraction based on N-FINDR is optimized using compute unified device architecture (CUDA) on GPUs as shown in Algorithm 2.

The NFINDR-GPU is graphically illustrated by a flowchart in Figure 1.
(1) Randomly initialize the set of \( q \) endmembers \( E^{(0)} \) from the dimensionality reduced hyperspectral data. \( E^{(0)} = \{ e_1^{(0)}, e_2^{(0)}, \ldots, e_q^{(0)} \} \), and \( k = 0 \);

(2) repeat

(3) Calculate the volume of the current endmember set \( E^{(k)} \)
\[
V(E^{(k)}) = \frac{1}{(q-1)!} \text{abs} \left( \begin{bmatrix} 1 \\ e_1^{(k)} \\ e_2^{(k)} \\ \vdots \\ e_q^{(k)} \end{bmatrix} \right)
\]

(4) For every pixel spectrum vector \( x_i \) (\( 1 \leq i \leq n \), \( n \) is the number of pixels) in the dimensionality reduced data set, test this pixel in all the \( q \) endmember positions in the \( E^{(k)} \) set and recalculate the volume \( V(E^{(k)}) \);

(5) If the new volume is larger than the previous one, then update the \( E^{(k)} \) set to \( E^{(k+1)} \), and \( k = k + 1 \), then go to step (3). If the new volume is not larger than the previous one, then no replacement is required;

(6) until the endmember set converges and no new replacement takes place.

Algorithm 1: N-FINDR.

(1) Randomly initialize the set of \( q \) endmembers \( E^{(0)} \) from the dimensionality reduced hyperspectral data. Set \( E^{(0)} = \{ e_1^{(0)}, e_2^{(0)}, \ldots, e_q^{(0)} \} \), and \( k = 0 \);

(2) Calculate the volume of the endmember set \( E^{(0)} \)
\[
V(E^{(0)}) = \frac{1}{(q-1)!} \text{abs} \left( \begin{bmatrix} 1 \\ e_1^{(0)} \\ e_2^{(0)} \\ \vdots \\ e_q^{(0)} \end{bmatrix} \right)
\]

(3) repeat

(4) Transmit the data set and the set of \( q \) endmembers \( E^{(k)} \) to the global memory of GPU;

(5) Calculation and Replacement: create \( n \) threads (\( n \) is the number of pixels in data set), each thread test a pixel in all the \( q \) endmember positions in the \( E^{(k)} \) and recalculate the volume \( V(E^{(k)}) \). Details of this step are listed below:

(5.1) set \( i = 1 \);

(5.2) repeat

(5.3) Create \( n \) threads for all the pixels;

(5.4) Each thread reads \( E^{(k)} \) from the global memory of GPU to their own registers respectively, and replace \( e_i^{(k)} \) in \( E^{(k)} \) with the corresponding pixel vector of the thread (GPU Kernel);

(5.5) Calculate the volume of the current endmember set \( E^{(k)} \), and store \( V(E^{(k)}) \) in the global memory (GPU Kernel);

(5.6) copy the volumes of all threads to internal memory, and find out the largest volume \( V_{\text{max}}(E^{(k)}) \);

(5.7) if \( V_{\text{max}}(E^{(k)}) > V(E^{(k)}) \), update \( k = k + 1 \), \( V(E^{(k)}) = V_{\text{max}}(E^{(k)}) \), and \( E^{(k)} = E^{(k)} \);

(5.8) \( i = i + 1 \);

(5.9) until \( i = q \);

(6) until the endmember set converges and no new replacement takes place.

Algorithm 2: Fast endmember extraction based on N-FINDR on GPUs (NFINDR-GPU).

4. Experiment

A well-known real hyperspectral scene labeled as f970619t01p02_r02, collected by the airborne visible infrared imaging spectrometer (AVIRIS) over the Cuprite mining district in Nevada [15], is used to evaluate the performance of NFINDR-GPU. In order to test the performance of the algorithm on different magnitude, the portions used in experiments correspond to 5 subsets of the scene, including a 250191-pixel subset, a 300300-pixel subset, a 350350-pixel subset (see Figure 2), a 512614-pixel subset, and a 2206614-pixel subset. The scene comprises 224 spectral bands between 0.4 and 2.5 \( \mu m \), with nominal spectral resolution of 10 nm. Prior to the analysis, bands 1–3, 105–115, 150–170, and 221–224 were removed due to water absorption and low SNR in those bands. The site is well understood mineralogically and has several exposed minerals of interest including alunite, buddingtonite, calcite, kaolinite, and muscovite [5]. Reference ground signatures of the above minerals are available in the form of a U.S. geological survey library (USGS) [16]. The number of endmember \( q \) was estimated to be 16 after calculating the VD of the AVIRIS Cuprite data.

The GPU platforms used to test our parallel algorithm are the NVidia Quadra 600 and Tesla C2050. The former features 96 processor cores, total dedicated memory of 1 GB, and memory bandwidth of 25.6 GB/s. The latter features
Randomly initialize the set of \( q \) endmembers \( E^{(0)} \), set \( k = 0 \)

Calculate the volume of \( E^{(0)} \)

Transmit the data set and \( E^{(k)} \) to the global memory of GPU

Set \( i = 1 \)

Create \( n \) threads

Each thread tests a pixel in all the \( q \) endmember positions in the \( E^{(k)} \) and recalculates the volume (GPU Kernel)

Find out the largest volume \( V_{\text{max}}(E^{(k)}) \)

\( V_{\text{max}}(E^{(k)}) > V(E^{(k)}) \)

Yes

\( k = k + 1 \), \( V(E^{(k)}) = V_{\text{max}}(E^{(k)}) \), \( E^{(k)} = E^{(k)} \)

No

\( i = i + 1 \)

\( i = q \)

Yes

\( E^{(k)} \) converges and no new replacement takes place

Yes

Get the endmember set

Figure 1: Flowchart of NFINDR-GPU.

448 processor cores operating at 1.15 GHz, with single precision floating point performance of 1030 Gflops, double precision floating point performance of 515 Gflops, total dedicated memory of 3 GB, 1.5 GHz memory (with 384-bit GDDR5 interface), and memory bandwidth of 144 GB/s. Both of the GPUs are connected to an Intel Xeon E5603 CPU at 1.6 GHz with 4 cores, with 8 GB RAM. The tests were performed on 64 b Microsoft Windows 7 operating system.

In order to achieve a fair benchmark in terms of execution time with respect to the GPU version, we firstly realize a C serial implementation of the N-FINDR algorithm as a basis for the subsequent parallel implementation. The serial algorithm is executed in one of the available CPU cores, and the parallel time is measured in the considered GPU platform. For each test, 60 runs are performed and the mean values are reported. The experimental results indicate that the endmember extraction results of both parallel and serial versions of N-FINDR correspond to the published ground truth very well. While using the same initialization endmember set \( E^{(0)} \), the serial and parallel algorithms reach the same maximum volume with the same iteration times. The main difference between them is the time they need to complete their calculations. The average run time per iteration and the speedup of the serial and parallel implementations are summarized in Table 1 and Figure 3.
Figure 2: A 350 × 350-pixel subset of the AVIRIS Cuprite scene.

Figure 3: The average speedup of the parallel implementations of N-FINDR.

Table 1: Average runtime and speedup of the serial and parallel implementations of N-FINDR.

| Data size  | Serial Time (sec) | Parallel_Quadro600 Time (sec) | Speedup (X) | Parallel_TeslaC2050 Time (sec) | Speedup (X) |
|------------|-------------------|-------------------------------|-------------|---------------------------------|-------------|
| 250 × 191  | 21.184            | 2.732                         | 7.75        | 0.675                           | 31.38       |
| 300 × 300  | 39.735            | 5.124                         | 7.75        | 1.245                           | 31.92       |
| 350 × 350  | 54.229            | 6.969                         | 7.78        | 1.698                           | 32.07       |
| 512 × 614  | 138.152           | 17.827                        | 7.75        | 4.308                           | 32.14       |
| 2206 × 614 | 596.901           | 76.735                        | 7.78        | 18.572                          | 32.14       |
We can conclude that the parallel implementations are stable for different image sizes, even for the big size of 2206 × 614 pixels, and NFINDR-GPU on Tesla C2050 achieved a significant speedup of greater than 30 times with regard to the CPU-based serial version of the N-FINDR algorithm. The proposed method shows better performance than the methods proposed in the existing literature. As [9] reports, while the AVIRIS scanning rate is 12 Hz, more recent satellite hyperspectral sensors such as Hyperion feature 220 Hz crossline scanning rates, which means that a hyperspectral sensor data like the AVIRIS Cuprite scene (a typical AVIRIS data cube with 614 × 512 pixels and 224 spectral bands) could be collected in about 5 s. The achieved processing time on the considered GPU architecture satisfies the real-time processing requirements.

5. Conclusions

Hyperspectral sensor can get the spectral signatures and enable identification of the materials that make up a scanned target, which will greatly improve the ability of target detection and recognition in sensor networks. Endmember extraction is one of the key issues for hyperspectral application in multisensor collaborative observation.

The N-FINDR is one of the most widely used and successfully applied methods for endmember extraction. But it suffers from long execution time due to its high algorithm complexity and the massive quantity of hyperspectral sensor data. High-performance computing is extremely demanded for those scenarios requiring real-time response, such as military detection, monitoring of chemical contamination, and wildfire tracking.

Improved parallel optimization of N-FINDR using the computer device unified architecture (CUDA) on GPUs (NFINDR-GPU) is proposed for fast endmember extraction of hyperspectral sensor data. The algorithm is implemented on both NVIDIA Quadra 600 and Telsa C2050, achieving significant improvements when compared with the previous GPU-based implementations of N-FINDR. Experimental results, based on the hyperspectral data collected by AVIRIS hyperspectral imaging sensor, show the effectiveness of NFINDR-GPU. The parallel implementation is stable for different image sizes, and the NFINDR-GPU on NVIDIA Tesla C2050 achieved a significant speedup of greater than 30 times with regard to the CPU-based serial version of N-FINDR, which satisfies the real-time processing requirements. Future work will focus on the comparison with other parallel methods.

Acknowledgments

Financial support for this work, provided by the National Natural Science Foundation of China (Grants nos. 61010194, 61071146), the Jiangsu Provincial Natural Science Foundation of China (Grant no. BK2011701), the Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20112219120024), the Jiangsu Province Six Top Talents project of China (Grant no. WLW-011), the CAST Innovation Foundation (Grant no. CAST201227), and the Project of China Geological Survey (Grant no. 1212011120227), is gratefully acknowledged.

References

[1] P. Lamborn and P. J. Williams, “Data fusion cm a distributed heterogeneous sensor network,” in Proceedings of the International Society for Optical Engineering (SPIE ’06), April 2006.
[2] R. M. Cavalli, L. Fusillli, S. Pasucci, S. Pignatti, and F. Santini, “Hyperspectral sensor data capability for retrieving complex urban land cover in comparison with multispectral data: venice city case study,” Sensors, vol. 8, no. 5, pp. 3299–3320, 2008.
[3] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente et al., “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. 354–3379, 2012.
[4] M. E. Winter, “N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data,” in Proceedings of the Imaging Spectrometry V (SPIE ’99), M. R. Descour and S. S. Shen, Eds., vol. 3753, pp. 266–275, 1999.
[5] S. Sanchez, R. Ramalho, L. Sousa, and A. Plaza, “Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs,” Journal of Real-Time Image Processing, 2012.
[6] A. Barberis, G. Danese, F. Leporati, A. Plaza, and E. Torti, “Real-time implementation of the vertex component analysis algorithm on GPUs,” IEEE Geoscience and Remote Sensing Letters, vol. 10, no. 2, pp. 251–255, 2013.
[7] C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B. Huang, “Recent developments in high performance computing for remote sensing: a review,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 4, no. 3, pp. 508–527, 2011.
[8] A. Plaza, Q. Du, Y.-L. Chang, and R. L. King, “High performance computing for hyperspectral remote sensing,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 4, no. 3, pp. 528–544, 2011.
[9] X. Wu, B. Huang, A. Plaza, Y. Li, and C. Wu, “Real-time implementation of the pixel purity index algorithm for endmember identification on GPUs,” IEEE Geoscience and Remote Sensing Letters. In press.
[10] A. Plaza, J. Plaza, and S. Sánchez, “Parallel implementation of endmember extraction algorithms using nvidia graphical processing units,” in Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS ’09), vol. 5, pp. V208–V211, July 2009.
[11] W. Luo, “Parallel implementation of N-FINDR algorithm for hyperspectral imagery on hybrid multiple-core GPU and GPU parallel platform,” in Proceedings of the Remote Sensing Image Processing, Geographic Information Systems, and Other Applications (SPIE ’11), November 2011.
[12] S. Sánchez, G. Martin, and A. Plaza, “Parallel implementation of the N-FINDR endmember extraction algorithm on commodity graphics processing units,” in Proceedings of the 30th IEEE International Geoscience and Remote Sensing Symposium (IGARSS ’10), pp. 955–958, July 2010.
[13] M. ElMaghraby, R. Ammar, and S. Rajasekaran, “Fast GPU algorithms for endmember extraction from hyperspectral images,” in Proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC ’12), pp. 631–636, Cappadocia, 2012.
[14] C.-I. Chang and Q. Du, “Estimation of number of spectrally distinct signal sources in hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 608–619, 2004.

[15] NASA Jet Propulsion Laboratory, Free AVIRIS Standard Data Products, 2013, http://aviris.jpl.nasa.gov/data/.

[16] U.S. Geological Survey, USGS Digital Spectral Library, 2013, http://speclab.cr.usgs.gov/spectral-lib.html.
Submit your manuscripts at
http://www.hindawi.com