FEATURE EXTRACTION OF HYPERSPECTRAL REMOTE SENSING IN PARALLEL COMPUTING RESEARCH BASED ON GPU

Luo Yaohua [a, *], Guo Ke [a], Zhao Shibo [a], Tao zhongping [b], Wang maozhi [a]

[a] Key Lab of Geomathematicas of Sichuan Province, Chengdu University of Technology, Chengdu, Sichuan Province, China- (lyh, guoke, zsb, wangmz)@cdut.edu.cn

[b] Information Technology Centre, Chengdu Sport University, Chengdu, Sichuan Province, China- 13588419@qq.com

KEY WORDS: GPU, parallel compute, PCA, hyperspectral remote sensing, feature extraction

ABSTRACT:

This paper introduces the basic modes of image parallel processing, and analyzes the parallel processing ability and stream program mode of GPU. At the same time, in view of the hyperspectral remote sensing data due to the large amount of data caused by the operational characteristics of a long time, study on feature extraction of hyperspectral remote sensing algorithms of parallel strategy, especially for the PCA parallel process based on GPU. At last, it makes a system experiment on hyperspectral remote sensing images to prove the validity of this method.

1. INTRODUCTION

As a new way of remote sensing, hyperspectral remote sensing has been in the military and civil fields play an important role in nearly 20 years, quickly and accurately dig out the required information from a large number of data is an important application.[1] The characteristics of hyperspectral remote sensing data are numerous channels, high spectral resolution, and large amounts of data, which makes It easy to discriminate objects in the scene, however, the vast amounts of data not only makes it difficult to transport and store them, but also process them. Therefore, it is very important to reduce the dimension in the hyperspectral image analysis. Dimensionality reduction can generally fall into feature extraction and band selection. As a multianalysis tool, Principle Component Analysis(PCA) is commonly used in feature extraction. But the PCA is a kind of complex algorithm, it's too time consuming to compute in serial mode with CPU.

The Graphics Processing Unit (GPU) has strong parallel processing ability and extremely high memory bandwidth. It has been abstract into a "stream processor" and used in fields that need a lot of repetitive data operation and intensive memory access, such as scientific calculation, data analysis, linear algebra and fluid simulation.

2. GPU PARALLEL PROCESSING TECHNOLOGY

The existing image processing algorithms are generally the serial algorithm, with the development of satellite remote sensing technology and progress, the resolution of remote sensing images are also constantly improving, data become more and more huge. When we handle the big data quantity image, the speed of serial processing is slowly, how to improve the processing speed is a problem that to be solved urgently. Image processing need large-scale matrix operations, so we can improve processing speed through the parallel processing. The existing parallel image processing method is stream parallel processing structure based on single instruction multiple data(SIMD),this method is concentrated at the bottom of the signal or image processing, its parallel particle size is small, and almost the parallel operation level. It is difficult to master and cannot meet the demand of complex tasks. Another method is stream parallel processing structure based on multiple instruction stream multiple data stream (MIMD), the typical MIMD system is composed by multiple processor, multiple memory blocks and an internet network, and each machine processor execute themselves' instruction processing data.

In 2007 the NVIDIA Company officially released the products - CUDA (Compute Unified Device Architecture), it's the first general development environment and software system that use class C language on the calculation, and it doesn't need to use graphics API. The basic idea of CUDA is to support a large number of thread level parallel, dynamic scheduling and execute these threads in hardware. For CUDA programming, GPU can be executed in parallel to multiple threads of computing devices and it can be operated as CPU coprocessor or host. It can put the parallel, calculation dense data from the applications that running on CPU into GPU device, the function that running in GPU is kernel functions. A complete program is composed of a number of kernel function and running on the CPU serial processing.

Fig.1 shows the difference between the CPU and GPU in micro-architecture.

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

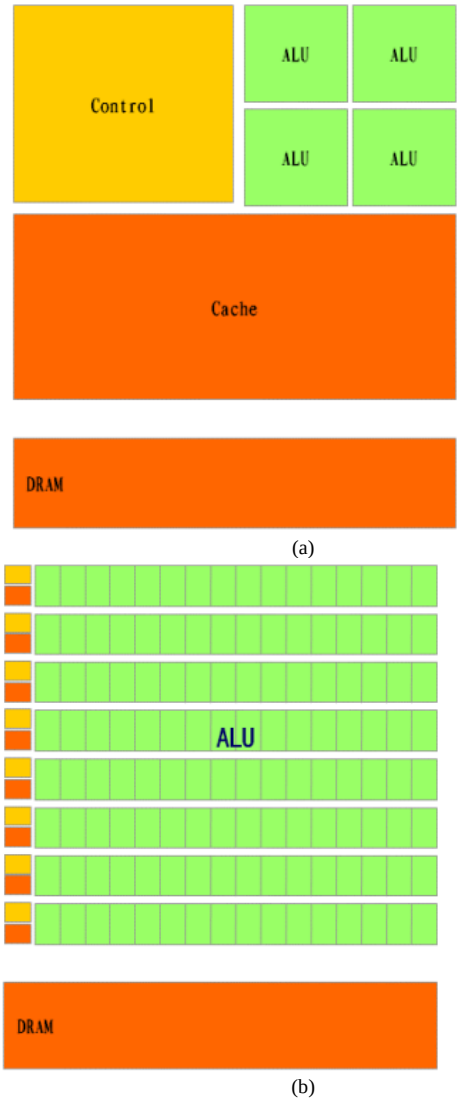

Figure 1. (a) the micro-architecture of CPU (b).the micro-architecture of GPU

Through three layers of hierarchy, CUDA can manage these threads. A certain number of thread composition thread pieces, and a certain number of threads block is composed of one-dimensional or two-dimensional thread block grid. The same block thread can cooperate with each other, through shared memory to share data, and its implementation to coordinate access to memory. A block of all threads must be located in the same processor core; the number of threads per block is limited by memory resources in processor core. A kernel function may be executed by the thread blocks which have the same size, the number of threads that execute the kernel function should be equal to each block's thread number plus the number of blocks, and we call these for thread block grid. If thread blocks needed to execute independently, it must be able to perform in any order whether parallel or sequential execution. The block number of thread in a grid is usually limited by the processing of the data size, rather than by the hardware processor number, the former may far exceed than latter in quantity. Fig.2 shows the thread block grid in GPU.

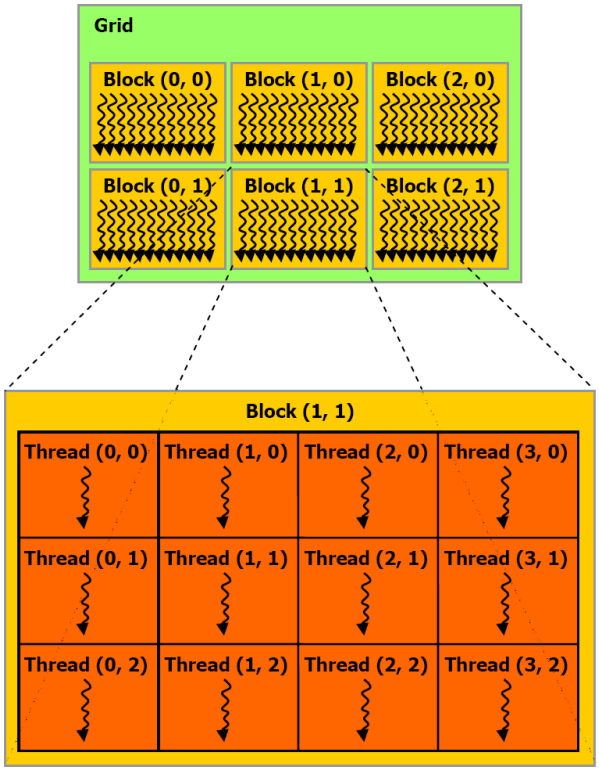

Figure 2. Thread block grid in GPU

3. INTRODUCTION OF PCA ALGORITHM

Feature extraction can compress the information of hyperspectral remote sensing image from high dimension to low dimension very fast. So it is widely used in dimension reduction of hyperspectral remote sensing.PCA is one of commonly used method. It extracts feature by multivariable linear transform, and generate low-dimensional spaces according to the size of signal-to-noise ratio. It eliminates the correlation basically in new space, and contains most of the original data information by several principle components. Compared with other feature extraction methods such as P, PC, ADA, DBFF, PCA calculates the global statistical characteristics. It is the transform with minimum mean square error.

PCA is a method to simplify data set, it is a kind of linear transformation to transform the data to a new coordinates which makes the biggest variance project to the first component and the second biggest variance project to the second component, and so on. PCA keeps the characteristics with the largest variance when reducing data set dimension by keeping low order principal component and ignoring high order principal component. Because the low order principal component usually content the most important data. $y_1, y_2, \cdots y_q (q < p)$

The algorithm is shown in detail as follows.

Suppose that $X = (x_1, x_2, \cdots, x_p)^T$ is a random variable with p dimensions, and $EX = u, DX = V \geq 0$, so there are p characteristic variables of the sample from a general X with average value u and covariance matrix V.

Suppose $y_1, y_2, \cdots y_q (q < p)$ is the linear combination of $X_1, X_2, \cdots, X_p$, and $y_1, y_2, \cdots y_q$ are irrelevant and

can reflect the information which can be reflected by p individual characteristic variables, so $y_1, y_2, \cdots y_q$ are named as the main component of $x_1, x_2, \cdots, x_p$. Here we take y_1 as example to explain the solve method of each main component.

It can be said that y_1 is composited by $x_1, x_2, \cdots, x_p$, for it can reflect the information sufficiently which is reflected by $x_1, x_2, \cdots, x_p$.

Suppose:

$$y_1 = a_1^T X = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p \quad (1)$$

Formula (1) can be obtained only by getting the value of vector a_1, i.e. how to choose an appropriate nonzero vector a_1 to make y_1 maximum reflect the information brought by former p individual characteristic variables, which is means that y_1 obtained by the alternate of $x_1, x_2, \cdots, x_p$ through formula (1) has maximum variance , and the value of formula (2) is as large as possible, where

$$Dy_1 = D(a_1^T X) \quad (2)$$

And formula (2) can be simplified as:

$$Dy_1 = a_1^T E[(X-u)(X-u)^T]a_1 = a_1^T DXa_1 = a_1^T Va_1 \quad (3)$$

There is no practical meaning to just make the variance of integrated variable y_1 as large as possible if no limitation is set to a_1 in formula (3). Generally, set a_1 a unit vector, i.e. $a_1 a_1^T = 1$.Lagrange multiplication should be employed to solve the conditional extremum of a_1 that make $Dy_1 = a_1^T Va_1$ reach the largest under the condition of $a_1 a_1^T = 1$.Set $\varphi(a_1, \lambda) = a_1^T Va_1 - \lambda(a_1^T a_1 - 1)$, where λ is the lagrange multiplier.

Solve partial derivatives of $\varphi(a_1, \lambda)$ about vector a_1 and multiplier λ respectively, and set it to zero, get

$$\begin{cases} \dfrac{\partial \varphi}{\partial a_1} = 2Va_1 - 2\lambda a_1 = 0 \\ \dfrac{\partial \varphi}{\partial \lambda} = -a_1^T a_1 + 1 = 0 \end{cases} \quad (4)$$

That is

$$\begin{cases} Va_1 = \lambda a_1 \cdots (1.4) \\ a_1^T a_1 = 1 \cdots (1.5) \end{cases}$$

Using a_1^T to premultiply formula (4), get

$$Dy_1 = a_1^T Va_1 = \lambda a_1^T a_1 = \lambda \quad (5)$$

Achieved by formula (4) again that

$$(V - \lambda I)a_1 = 0 \quad (6)$$

If make equations (6) have untrivial-solution, the necessary and sufficient condition is

$$|V - \lambda I| = 0 \quad (7)$$

Thus, λ is also the characteristic root of covariance matrix V. From formula (6) it can be seen that to get the largest value of Dy_1, we should make λ have the largest value. So λ is equal to the maximum characteristic root λ_1 of covariance matrix V, and a_1 is the unit characteristic vector corresponding to λ_1. Thus the first multiple variable is solved.

$$y_1 = a_1^T X \quad (8)$$

And y1 is named as the first main component.

4. PARALLEL ALGORITHM OF PCA

According to the principle of PCA, we know that it is mainly concentrated in the process of calculating the covariance matrix, so the parallel algorithm of PCA is mainly concentrated in calculation of the covariance matrix. Fig3 shows the optimization chart.

Read spectrum library data
Calculate covariance
Calculate feature vector
Sort according to the variance
Calculate principal component
Transform the original data according to the principal component

Figure 3. Optimization chart

The parallel algorithm is shown in detail as follows.

The formula of covariance matrix is

$$\sigma_{xy} = COV(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{S-1} \quad (9)$$

Where X,Y= the band

S= number of each band

Transform the formula (9) as fellow,

$$COV(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{S-1}$$

$$= \frac{1}{S-1}\left[\sum_{i=1}^{s} X_i Y_i + \overline{X}\overline{Y} - X_i\overline{Y} - \overline{X}Y_i\right]$$

$$= \frac{1}{S-1}\left[\sum_{i=1}^{s} X_i Y_i - S\overline{X}\overline{Y}\right]$$

$$= \frac{1}{S-1}\left[\sum_{i=1}^{s} X_i Y_i - \frac{1}{S}\sum_{i=1}^{s} X_i \sum_{i=1}^{s} Y_i\right]$$

$$= \frac{1}{S-1}\begin{bmatrix} (\sum_{1} X_i Y_i + \sum_{2} X_i Y_i + \cdots \sum_{n} X_i Y_i) - \\ \frac{1}{S}(\sum_{1} X_i + \sum_{2} X_i + \cdots \sum_{n} X_i) \\ (\sum_{1} Y_i + \sum_{2} Y_i + \cdots \sum_{n} Y_i) \end{bmatrix} \quad (10)$$

Where n = the number of nodes.

Support the number of original remote sensing band is n, the covariance matrix is V , It will make n^2 GPU processes, and each process calculates the component of each V .

5. RESULT AND ANALYSIS

The hardware platform used in this experiment is: cpu: Intel(R) CoreI7-2630QM 2.0GHZ; memory size: 4GB; video card: GeForce GT540M2GB; OS: Windows7 Professional Edition 64; development environment: Microsoft Visual studio 2008; 3D interface: OpenGL. The hyperspectral remote sensing data respectively with CPU serial and GPU parallel computing,The result shows that the parallel PCA has obvious acceleration effect based on GPU, especially in the larger amount of data processing, the effect is more obvious, the value is more than 4 times. Fully display the advantageor computationally intensive problems in GPU parallel computing.

6. CONCLUSIONS

Hyperspectral remote sensing data processing often involves large amount of data process and complexoperation. With the rapid development of GPU, we can simplify the process of design, making full use of powerful floating-point arithmetic ability in stream processor unit to solve complex scientific computing problems. This has brought a new change to solve a large amount of data processing. We can find obviously advantage in calculation speed by use GPU parallel computing, and this mode can be used in other aspects of the processing in hyperspectral remote sensing. It will be the future development trend of hyperspectral remote sensing.

7. ACKNOWLEDGMENT

This paper is supported by the 863 project (2008AA121103), and the Natural Science of Education Department of Sichuan Province project (09ZA004), China Geological Survey projects (1212011120226, 1212011120227, 1212011087084), open funds of Geomathematics Key Lab of Sichuan Province (SCSXDZ2009004)

8. REFERENCES

References from Journals:

YANG Jing-yu,ZHANG Yong-sheng, DONG Guang-jun.Investigation of parallel method of RS image SAM algorithmic based on GPU. *Science of Surveying and Mapping*,2010(5),pp.9-11

LU Jun, ZHANG Bao-ming, HUANG Wei, LI Er-sen.IHS Transform Algorithm of Remote Sensing Image Data Fusion Based on GPU. *Computer Engineering*,2009,7(35),pp.261-263.

HE Zhong-hai,HE Bin-bin.Weight Spectral Angle Mapper(WSAM) Method for Hyperspectral Mineral Mapping. *Spectroscopy and Spectral Analysis*,2011(8)

Svetlin A ,Manavski ,Giorgio Valle. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment.*BMC Bioinformatics* , 2008,9(2),pp.S10-2008.

Song ji,Zhou song-tao.Fast Image Matching Based on GPU Parallel Computing.*Journal of Hubei University for Nationalities(Natural Science Edition)*,2011,(9),pp.306-310

References from Books:

Jason Sanders Edward Kandrot, *CUDA by Example:an Introduction to General-Purpose GPU Programming*,2011,1

Hubert Nguyen, *GPU GEMS3*, 2011, 9

Chen Xian-feng,Warner Timothy A,Campagna David J.*Remote Sensing of Enviroment,2007*,

Wang jin-nian,Zheng lan-fen,Tong qing-xi.*Romote Sensing of Environment china*,1996,11