ACCELERATING INSAR RAW DATA SIMULATION ON GPU USING CUDA

Zhang Fan[1], *Wang Bing-nan*[2], *Xiang Mao-sheng*[2]

1．College of Information Science & Technology, Beijing University of Chemical Technology, Beijing, 100029, China；2．National Key Laboratory of Microwave Imaging Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing, 100190, China

ABSTRACT

This paper describes a scalable parallel method for interferometric synthetic aperture radar (InSAR) raw data simulation on graphic processing unit (GPU) with common unified device architecture (CUDA). The advantages of the new method rely on the three contributions: GPU hardware provides lots of stream processors for threads calculating, CUDA software environment runs thousands of threads working in parallel for assigned task, raw data simulation adopts the fine-grained task parallelism. Compared with OpenMP, MPI and grid computing, the method not only improves the computational efficiency greatly, but also save the resources such as hardware, electric power and room space. The results show that the method not only ensures accuracy, but also be able to obtain the speedup about 30 times.

Index Terms— Interferometric synthetic aperture radar, raw data generation, parallel processing, simulation

1. INTRODUCTION

As the image processing algorithms is relatively mature, the main task of InSAR system simulation is to simulate the raw data that contains different actual system errors. With the increased resolution of InSAR systems and the demand for InSAR applications, the objects of InSAR echo simulation change from surface target into nature three-dimensional terrain, which brings the rapid growth of computation time.

So the OpenMP, MPI, grid computing and other high-performance parallel computing technology are used to optimize and accelerate the simulation courses [1-3]. The three kinds of parallel computing technologies are based on CPU platform, which is designed for the control and logic processing. This design pattern will cost more in construction and maintenance for large-scale parallel computing.

As the improvement of GPU parallel processing capacity and programmability, the combination of GPGPU (general purpose GPU) technology and the classical algorithm has been a hot topic at present GPGPU has been applied to biology, geophysics, medical imaging, image processing and other areas [4-7]. Compared with CPU it achieves acceleration of a dozen to several hundred times. Therefore the paper presents the InSAR raw data parallel simulation algorithm on GPU using CUDA. The analysis of computational efficiency and effect proves that the GPU acceleration is valid, and can be applied to InSAR simulation, InSAR data processing, InSAR image application and so on.

The next Section introduces the basic of CUDA. Section 3 analyzes the Parallelizability of SAR raw data simulation and describes the parallel method with CUDA. Then, Section 4 illustrates the speed and imaging result of the method. Finally, conclusions are drawn in Section 5.

2. COMPUTE UNIFIED DEVICE ARCHITECTURE

GPGPU apply graphics operation to the general non-graphics computing. It increases computation speed greatly with a lot of data processing units. But the API interface of OpenGL and DirectX is needed to change non-graphic description to graphical solution in the traditional GPGPU application. It causes great inconvenience to the researchers.

CUDA that introduced by NVIDIA corporation is a new parallel programming model and software environment, which solve the above-mentioned inconvenience in GPGPU computing applications [11]. CUDA architecture uses the syntax similar to C language and function library instead of the operations in the graphics API. GPU program is developed more flexibly with CUDA, which brings about wide application in various fields. When the program runs, GPU works as a co-processor of CPU and runs thousands of threads working in parallel to achieve acceleration.

Compared with the hardware architecture of CPU and GPU, the majority transistors in CPU is used to control and buffer, while the majority transistors in GPU is designed for data processing. The difference in hardware design determines the magnitude of their floating-point computing power. Currently NVIDIA Tesla S1070 has 4T Flops computing ability and 408G/s peak bandwidth, which far exceed popular CPU. Therefore, GPU solution is particularly suitable for data parallel computing, high-density computing problems.

3. SCALABLE PARALLEL INSAR RAW DATA SIMULATION

The echo signal model by Wu C. [12] is applicable for airborne, spaceborne SAR echo signal model. Assuming the transmitting pulse is a linear frequency modulated signal $s_r(\tau)$:

$$s_r(\tau) = rect(\frac{\tau}{T_p})\exp(jw_c\tau + j\pi k_r\tau^2) \tag{1}$$

Through coherent receiving the single point echo is expressed as two-dimensional $s(t, \tau)$:

$$\begin{aligned} s(t,\tau) = \sigma W_a(\theta) rect(\frac{t}{T_a})\exp(-j\frac{4\pi r(t)}{\lambda}) \\ rect(\frac{\tau - \frac{2r(t)}{c}}{T_p})\exp(j\pi k_r(\tau - \frac{2r(t)}{c})^2) \end{aligned} \tag{2}$$

Where t is the azimuth time, τ is the range time, σ is the scattering coefficient, W_a is the antenna gain, θ is the antenna look angle, T_p is the signal pulse width, T_a is the synthetic aperture time, $r(t)$ is the distance between target point and the radar antenna phase center at time t, k_r is the signal modulation frequency, rect (.) is a rectangular envelope.

When the simulation objects are distributed targets, the SAR echo signal can be obtained as (3):

$$\begin{aligned} s(t,\tau) = \sum_{n=0}^{T}\sum_{i=1}^{M}\sigma_{0i}W_a(\theta_i)s_r(\tau - \frac{2r_i(t_n)}{c}) \\ rect(\frac{t_n}{T_a})\exp(-j\frac{4\pi r_i(t_n)}{\lambda}) \end{aligned} \tag{3}$$

Where i is the order number of distributed points in scattering matrix, n is order number in azimuth time.

In practical engineering calculation, the improved algorithm [13] of time-domain pulse coherent method is often applied, the principles as shown in (4):

$$\begin{aligned} s(t,\tau) &= \sum_{n=0}^{T} s_a(t_n,\tau)\otimes s_r(\tau) \\ &= \sum_{n=0}^{T} f^{-1}\{f[s_a(t_n,\tau)]\ S_r(\xi)\} \end{aligned}$$

with

$$\begin{aligned} s_a(t_n,\tau) = \sum_{i=1}^{M}\sigma_{0i}W_a(\theta_i)\exp(-j(\frac{4\pi r_i(t_n)}{\lambda})) \\ \delta(\tau - \frac{2r_i(t_n)}{c}) \end{aligned} \tag{4}$$

Where $f(.)$ is the Fourier transform operator, $f^{1}(.)$ is inverse Fourier transform operator, $S_r(\xi)$ is the linear FM Signal spectrum.

In the course of simulation, the linear FM signal spectrum $S(\xi)$ is changeless, while the azimuth signal spectrum is changed with different scattering point and azimuth time. Accordance with (4), the raw data simulation algorithm includes five steps:

1) The linear FM signal spectrum is calculated;
2) The azimuth signals of all scattering points are calculated, and transformed into the frequency domain;
3) The multiplication of azimuth signal and linear FM signal spectrum is completed;
4) The raw data of one channel is achieved by the inverse Fourier transform of results.
5) For the other channel 2)-4) is repeated to get the raw data.

For all the azimuth time samples, 1) - 5) is repeated to get the whole simulated raw data pair.

According to stop-and-go model, SAR/InSAR raw data simulation is a time sequence course, and the coupling of transmitting and receiving pulses at different times is small. So you can take the transmitting and receiving pulses as the task, which will be dispatched to every computation node and calculated quickly by GPU, MPI, grid computing or other parallel technologies.

The GPU-based InSAR raw data parallel simulation runs a large number of GPU threads for parallel computation. Each GPU thread calculates the echo of single scattering point at each azimuth time. Although the computing power of each GPU thread is weaker than CPU, the scalable parallel computation maximizes the GPU hardware floating-point operation capacity, and improve the computing speed about 1-2 orders of magnitude finally.

To make use of obvious advantage of GPU parallel computing, the InSAR raw data parallel simulation algorithm on GPU using CUDA is proposed, and the algorithm flow of one channel raw data simulation is as follows:

Step I According to paper[8-10], 3D target is placed in a flat rectangle to build scene geometry model firstly. Secondly geometric characteristics map is calculated using GPU automatic blanking. Thirdly the scattering coefficient map of 3D target is calculated using GRECO algorithm.

Step II According to the system parameter, the linear FM signal is calculated, converted to frequency domain, and transmitted to the GPU memory;

Step III Geometric characteristics map, scattering coefficient maps and geometrical parameters are transferred to the GPU memory and used to calculate the azimuth signals of all the scattering points by CUDA kernels;

Step IV The azimuth signal spectrum is calculated by CUDA CUFFT library. Then the spectrum multiplication is operated by the CUDA kernels. Finally the whole echo data is achieved by transforming the spectrum product to time domain using the CUFFT inverse Fourier transform. After that, (I) - (IV) are repeated until the end of simulation time. So repeating the course, we can get the other channel InSAR raw data by different geometrical parameters with the algorithm flow.

4. RESULTS

Through the above introduction of CUDA, the technology can be introduced into InSAR raw data simulation algorithm for large-scale parallel computation. The following airborne InSAR raw data simulation experiments are designed to analyze the speedup and imaging effects. The Intel Core2 Quad Q6600 (2.4GHz) CPU, NVIDIA Geforce 9500GT GPU and CUDA Toolkit 2.0 are used in the experiments, whose simulation parameters are shown in Table I.

TABLE I
Simulation Parameters

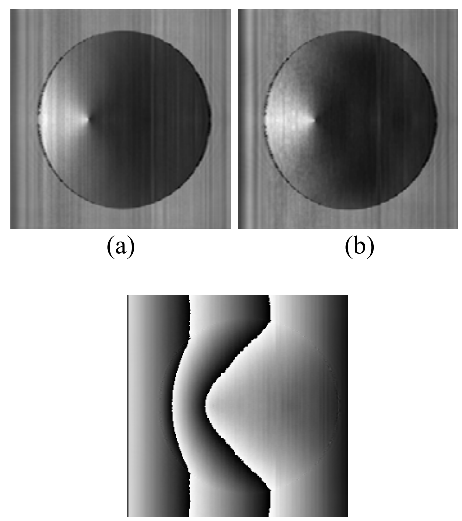
Parameters	Value
Baseline	1m
Wave length	0.05m
PRF	300Hz
Pulse width	10us
Band width	30MHz
Sampling rate	60MHz
Look angle	45deg
Flight velocity	130m/s
Center distance	10000m
Data size	1024x512
Scene size	100x100

The three-dimension targets simulation is needed in the InSAR simulation and applications. So a simple analysis of speed acceleration in three-dimension simulation will be given in the section. The target elevation information is added in three-dimension target raw data simulation. The extra dimension makes the calculation of azimuth signal slant range and look angle increase. Therefore, the simulation speed of three-dimension cone and two-dimension flat surface should be compared under the same conditions. In the experiment, the target area size is 100x100. The imaging results of simulated echo pair is shown in Figure 1, and the speed experiment and results are shown in Table II.

TABLE II
Simulation Speed Comparison

Target	CPU	GPU	Speedup
3D	20613ms	703ms	29.32
2D	9531ms	411ms	23.19

As can be seen in Table II, speed acceleration in three-dimensional simulation is more obviously. Compared with two-dimensional raw data simulation, the speedup increases about 6 times. It can be seen that the SAR raw data fine-grained parallel simulation with CUDA is more suitable for simulation of three-dimensional targets, and easy to apply to InSAR simulation.



(a) (b) (c)

Figure 1 Imaging results of simulated 3D cone in different channel (a, b) and their interferometric fringe (c)

5. CONCLUSION

The GPU general purpose computing has been more widely used with the development of CUDA architecture. The paper introduces CUDA technology to solve calculation bottleneck of InSAR raw data simulation. The experiments results show that the method is 29 times faster than CPU methods, and have advantages of small space occupier, low electric power consumption and low hardware cost. The method is especially suitable for airborne / spaceborne SAR simulation of large-scale scenes and InSAR simulation. The next step of research will apply CUDA parallel computing technology to spaceborne InSAR echo simulation of real terrain, which assists the spaceborne InSAR system design and validation.

6. REFERENCES

[1] Su Yu, Qi Xiang-yang. OpenMP based space-borne SAR raw signal parallel simulation. *Journal of Graduate School of the Chinese Academy of Sciences*, 2008, 25(1):129～135.

[2] Wang Xi-shuang, Huang Li-sheng, Wang Zhen-song. Research on parallel arithmetic of distribute space borne SAR ground target simulation. *Journal of System Simulation*, 2006, 18(8):2097～2100.

[3] Zhang Fan, Lin Yin, Hong Wen. SAR echo distributed simulation based on grid computing. *Journal of System Simulation*, 2008, 20(12):3165～3170.

[4] Derek Juba, Amitabh Varshney. Parallel, stochastic measurement of molecular surface area. *Journal of Molecular Graphics and Modeling*, 2008, 27:82～87.

[5] Li Bo, Liu Guo-feng, Liu Hong. A method of using GPU to accelerate seismic pre-stack time migration. *Chinese Journal of Geophysics (in Chinese)*, 2009, 52(1):245～252.

[6] Scherl H, Keck B, Kowarschik M, et al. Fast GPU-Based CT Reconstruction using the Common Unified Device Architecture (CUDA). *NSS'07*, 2007, 6:4464～4466.

[7] Yang Zhi-yi, Zhu Ya-ting, Pu Yong. Parallel Image Processing Based on CUDA. *2008 International Conference on Computer Science and Software Engineering*, 2008, 3:198～201.

[8] Zhang Fan, Hong Wen. Geometrical Distortions Simulation of SAR Image Base on Computer Graphics. *Joural of System Simulation*, 2009, 21(9):2503～2508.

[9] Fan Zhang, Lu Bai, Wen Hong. InSAR Imaging Geometry Simulation Based on Computer Graphics. *ISPRS 2008*, 2008, XXXVII(B1):781～784.

[10] Fan Zhang, Wen Hong, Daojing Li. SAR Image Simulation of Man-Made Scenes Based on Computer Graphics. *IGARSS 2008*, 2008, IV:1395～1397.

[11] NVIDIA Corporation. “CUDA Programming Guide 2.0”. http://www.nvidia.com, 2009.

[12] Wu C, Liu K Y, Jin M, “Modeling and a correlation algorithm for spaceborne SAR signals”, IEEE Trans. on Aerospace and Electronic Systems, vol.18, no.5, pp.563–574, 1982.

[13] Huang Li-sheng, Wang Zhen-song, Zheng Tian-yao, “A fast algorithm based on FFT used in simulation of SAR return wave signal”, Journal of Remote Sensing, vol.8, no.2, pp.128–136, 2004.