International Journal of Physical Sciences Vol. 7(8), pp. 1224 - 1234, 16 February, 2012
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.1625
ISSN 1992 - 1950 © 2012 Academic Journals

Full Length Research Paper

Hybrid general-purpose computation on GPU (GPGPU) and computer graphics synthetic aperture radar simulation for complex scenes

Fan Zhang[1]*, Zheng Li[1], Bingnan Wang[2], Maosheng Xiang[2] and Wen Hong[2]

[1]College of Information Science and 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.

Accepted 13 January, 2012

In this paper, a new hybrid general-purpose computation on GPU (GPGPU) and computer graphics synthetic aperture radar (SAR) simulation method for complex scenes is proposed. Previous SAR simulations for complex scenes only use GPU's graphics capabilities for scattering calculation in graphical electromagnetic computing (GRECO) algorithm. The new hybrid method use GPU's graphics and parallel computing capabilities for geometry modeling, scattering map and raw data calculation in SAR simulation of complex scenes. The advantages of the new method rely on the three contributions: GPU hardware provides lots of stream processors for threads calculating, common unified device architecture (CUDA) environment runs thousands of threads working in parallel for assigned task, raw data simulation adopts the fine-grained task parallelism. Compared with classical algorithms, the method not only ensures the accuracy of scattering calculation with GRECO algorithm, but also improves the computational efficiency greatly for complex scenes consideration. The results show that the method is able to obtain the speedup about 30 times on entry-level GPU.

Key words: Synthetic aperture radar (SAR), raw data simulation, parallel processing, general-purpose computation on GPU (GPGPU), computer graphics.

INTRODUCTION

With the development of microwave remote sensing and people's demand for high-resolution Synthetic aperture radar (SAR) images, more air-borne/space-borne SAR systems will be developed in the future. Due to the difficulty and high-cost of SAR flight experiments, computer simulation is often used to assist the research of key technologies, system design and system development.

The SAR raw data simulation with complex scenes and various actual system errors are generally used to assist the imaging algorithm research, radar system design, system verification and so on. With the increased resolution of SAR systems and the demand for image interpretation, the objects of SAR raw data simulation change from point target, surface target to three-dimensional complex target, which brings the challenges to targets modeling, scattering calculation and raw data simulation. To adequately solve the two former problems, computer graphics (CG) (Zhang et al., 2008A; Zhang et al., 2008B) and graphical electromagnetic computing (GRECO) (Margarit et al., 2007) all use GPU hardware to accelerate the geometric processing which is introduced into SAR simulation. In addition, GPU are used to research more complex backscattering mechanism, such as the multi-bounce effects (Timo and Uwe, 2009).

The high resolution and wide swath brings the rapid growth of SAR raw data computation time, even up to several days. To solve the problems better, algorithm research and high performance computing (HPC) should

*Corresponding author. E-mail: zhangf@mail.buct.edu.cn.

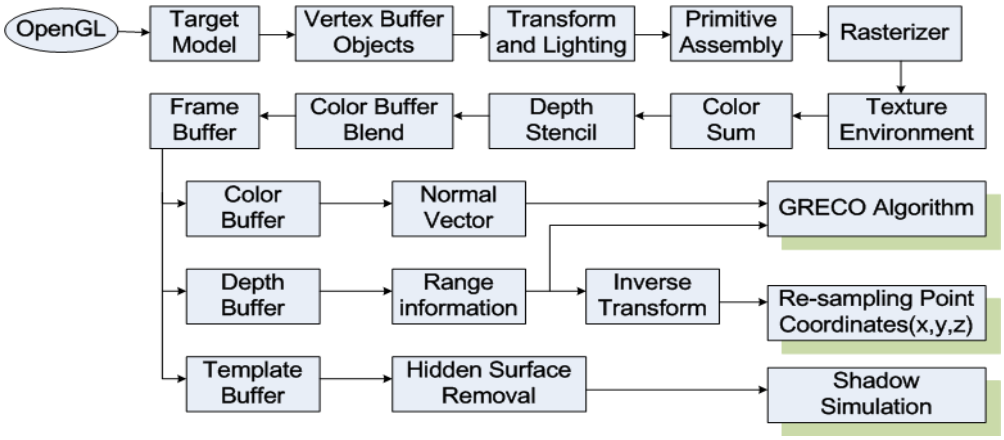


Figure 1. Computer graphics based method for geometry and scattering calculation.

be the two feasible solutions. Currently, raw data simulation algorithm is divided into three categories, including time domain pulse coherent algorithm, frequency domain pulse coherent algorithm (Liu, 1999), two-dimensional frequency-domain algorithm (Franceschetti et al., 1992). Two-dimensional frequency domain method is the fastest, but it is difficult to add the actual system errors. Although the former two kinds of algorithms can properly consider the actual system errors, the time consuming problem is quite severe. So the OpenMP (Su and Qi, 2008), message passing interface (MPI) (Wang et al., 2006), grid computing (Zhang et al., 2008c) and other high performance computing technology are used to optimize and accelerate the simulation courses. These three parallel computing technologies are based on CPU platform, which is designed for the control and logic processing. This design pattern will cost more resources in construction and maintenance for large-scale parallel computing. Since the improvement of GPU parallel processing capacity and programmability, the combination of GPGPU technology and the classical algorithm has been a serious topic in recent times (Garland et al., 2008; Zhang F et al. 2010). GPGPU has been applied to biology, geophysics, medical imaging, image processing and other areas. Compared with CPU, it achieves acceleration of a dozen to several hundred times.

In this paper, we proposed the hybrid GPGPU and computer graphics method to ensure both efficiency and accuracy of complex scenes SAR raw data simulation, which can also be applied to interferometric SAR, bistatic SAR, multiple-input multiple-output (MIMO) SAR simulation.

COMPUTER GRAPHICS BASED GEOMETRY AND SCATTERING SIMULATION

This method utilizes the GPU's fixed graphics pipeline, and applies to the SAR echo simulation process of geometry modeling, scattering calculation. The GPGPU development consists of three stages: The first stage is fixed graphics pipeline technology; the second stage is programmable graphics pipeline technology, and the third stage is CUDA scalable parallel computing technology. The former two stages all need to use OpenGL and other graphics languages to describe general-purpose computing issue with graphics calculation, which is very difficult for researchers, as shown in Figure 1. The last stages use CUDA C language interface to describe general-purpose computing issue, which is easier for researchers.

The CG based simulation method is based on fixed graphics pipeline with OpenGL or other graphic languages on GPU. The classical method generally includes three steps:

The first step is geometry process. Through the OpenGL operations, the vertex information of target model was acquired. Then the vertex information was processed by transformation and lighting. In this step, the vertex information of target is changed from object coordinate to screen coordinate.

The second step is rasterization. Through the process of texture, color and depth, the assembly element information was changed into pixels on the screen. All the pixels information was written into frame buffer.

The third step is buffer operation. The template buffer can be used for hidden surfaces removal. The shadow computation in geometry modeling will be finished by this way. The target normal vector can be acquired from color buffer. The range information can be acquired from depth buffer. Through inverse transformation, the re-sampling target points coordinates can be obtained after hidden surface removal, and then provided to the geometry calculation. With the normal vector and range information, the scattering map is calculated using GRECO, which is

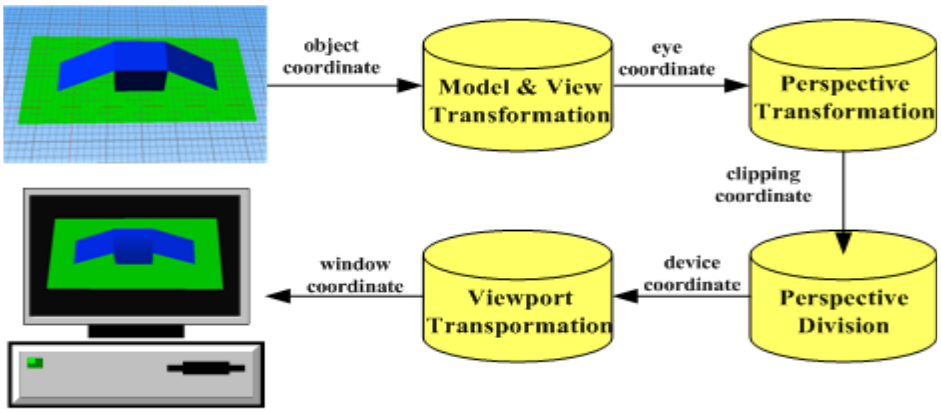


Figure 2. Graphic transformation.

Figure 3. Scene model rasterization.

equal to CG plus high frequency approximation RCS calculation method.

Geometry simulation

The geometric features of SAR image includes shadow, layover and foreshortening, which are the special characteristics compared with optical image and should be considered in SAR raw data simulation. In the course of traditional geometric features simulation, two problems are more important, one is interpolation algorithms which are used to deal with the target resampling in slant distance coordinates, and the other is shadow computation. The interpolation algorithms can introduce errors by interpolating functions; the shadow computation consumes much time for point-by-point comparison. To enhance the computational efficiency and precision of traditional geometric features simulation, CG method was introduced. The hidden surfaces in CG and the shadow in a SAR image are essentially the same; all of them are inaccessible regions due to sight or microwave obstructions. So the hidden surfaces removal in CG will be used to determine the shadow region in SAR simulated scene. Otherwise, the pixel operation can get enough sampling points without interpolation method.

The CG based simulation method includes three main steps:

First, the 3D model of scenes will be completed. The real terrain is simplified to a flat target, so the scene model can be constructed by placing the targets on a flat in CAD software.

Second, the scene model will be shown on the computer screen using CG process based on OpenGL, as is shown in Figures 2 and 3. The shadow area (hidden faces) will be removed by z-buffer algorithm. In the application of OpenGL, the z-buffer processing is automatically executed by the GPU card, which guarantees the simulation precision and efficiency.

Third, the size of the displayed target on screen should be adjusted for that the pixel numbers of target on screen is same as the numbers of sampling points in simulation.

The three-dimension geodetic coordinates of visible sampling points will be obtained through pixel operation, graphics transformation and inverse transformation, as shown in Figure 4.

Finally, the 3D model is changed into a heap of dense

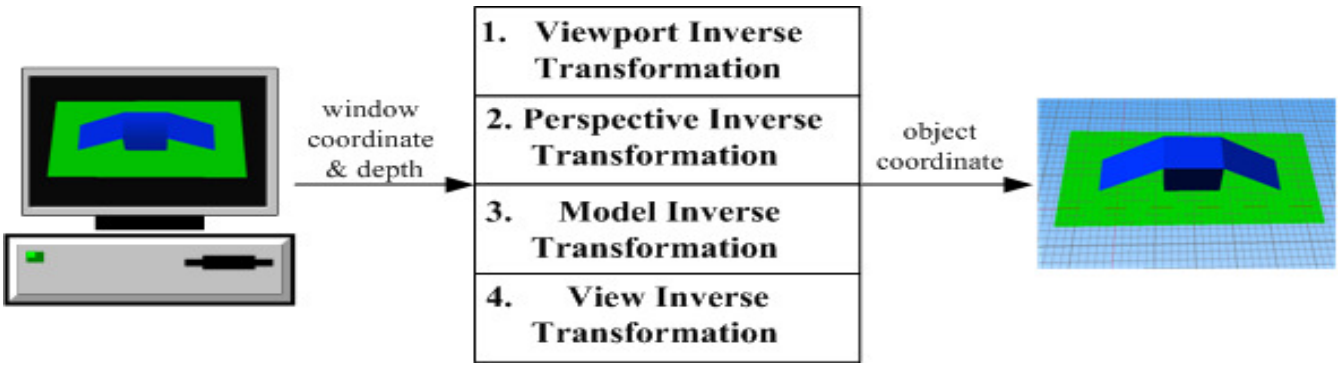


Figure 4. Graphic inverse transformation.

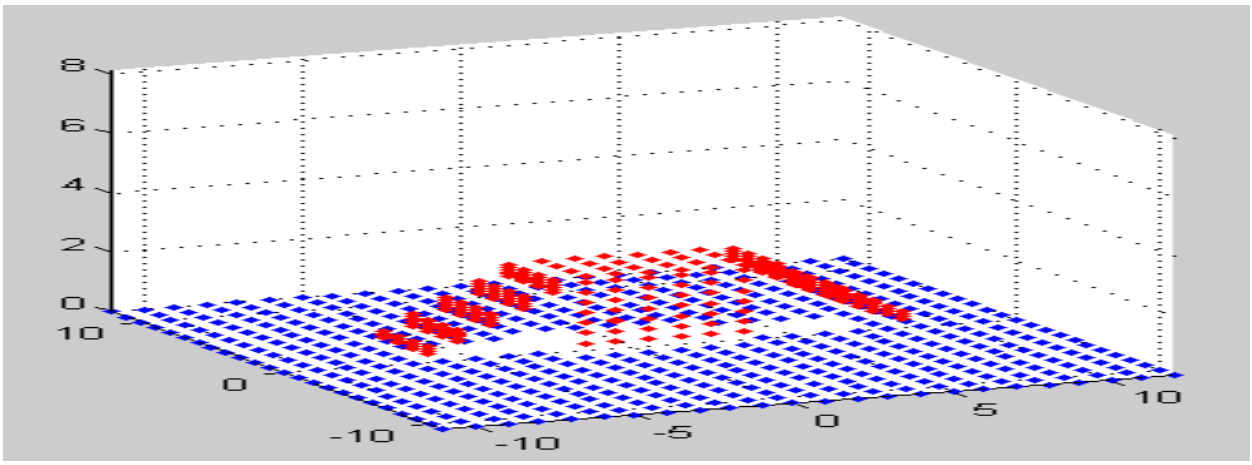


Figure 5. Re-sampling of scene model with CG process.

3D point array, which will be mapped to slant distance coordinates after resampling, as is shown in Figure 5.

Scattering simulation

GRECO method is regarded as one of the most effective method for high frequency approximation RCS calculation, so it is used to calculate the scattering map of complex target (Rius et al., 1993). The application of GRECO in SAR is easier than traditional RCS calculation for its long distance and narrow scope of look angle, so the scattering map can be considered unchanged in time and space (Margarit et al., 2006). In the scattering simulation, the surfaces scattering will be calculated by physical optic method; the diffraction of edge and vertex will be neglected in the simulation. The scattering map simulation includes two main parts:

Firstly, one light source should be placed on the location of the observer, and the scene needs to be rendered using the Phong local illumination model, as is shown in Figure 6a. The color and depth value of each pixel can be obtained to calculate its RCS using PO expression, where color value indicate the normal vector and depth value indicate the distance between source and target, as is shown in Figures 6b to c. And then the scattering map of targets and flat terrain will be calculated using Equation 1. The calculation result is shown in Figure 6. Moreover, the random phase of RCS should be added to every point in the scene.

$$\sigma = \frac{4\pi}{\lambda^2}\left|\sum_{PIXELS}\cos^n\theta\,\sin c\left(\frac{kl}{\cos\theta}\sin\theta\right)\exp(2jkz)\Delta s\right|^2 \quad (1)$$

where σ is the monostatic RCS of object surface, λ is the wavelength of electromagnetic waves, n is a parameter that controls the effect of this stationary phase approximation, k is the wave number, Δs is the surface

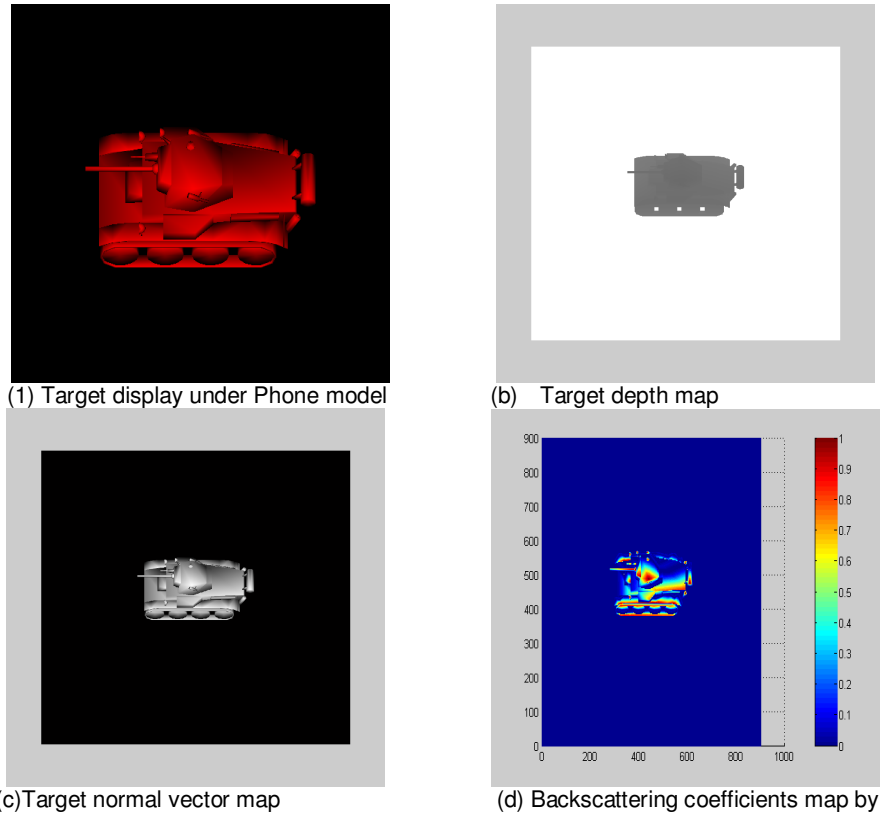


Figure 6. Intermediate results in GRECO.

unit of object, θ is the angle between the normal to the surface and the direction of incidence (derived from color value), z is the range between every pixels and the radiation source (derived from depth value) and l is the size of a square pixel in the screen.

Secondly, the scattering map of flat terrain should be replaced by optical images, airborne SAR images, and other remote sensing images. The pixel amplitude of the relevant images can be used as the backscattering coefficients of scenes. According to the height information of target points that is calculated previously, the replacement of SAR image to scattering map on flat terrain can be finished quickly. Before the replacement, the scattering map should compare with the SAR image in numerical value, and the quantization range of scattering map should be adjusted to match the gray level of SAR image. In the end, the RCS of All the target points besides shadow and interested target are replaced with SAR image gray values.

GPGPU BASED RAW DATA SIMULATION

CUDA base GPGPU

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, introduced by NVIDIA Corporation is a new parallel programming model and software environment, which solves the aforementioned inconvenience in GPGPU computing applications (Gray, 2009). 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.

SAR raw data simulation algorithm

The echo signal model by Wu et al. (1982) is applicable for airborne, spaceborne SAR echo signal model. Assuming the transmitting pulse is a linear frequency modulated signal $s_t(\tau)$:

$$s_t(\tau) = rect(\frac{\tau}{T_p})\exp(jw_c\tau + j\pi k_r\tau^2) \quad (2)$$

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

$$s(t,\tau) = \sigma W_a(\theta) rect(\frac{t}{T_a}) \exp(-j\frac{4\pi r(t)}{\lambda}) \cdot rect(\frac{\tau - \frac{2r(t)}{c}}{T_p}) \exp(j\pi k_r(\tau - \frac{2r(t)}{c})^2) \quad (3)$$

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, and $rect(.)$ is a rectangular envelope.

When the simulation objects are distributed targets, the SAR echo signal can be obtained as Equation 4:

$$s(t,\tau) = \sum_{n=0}^{T}\sum_{i=1}^{M}\sigma_{0i}W_a(\theta_i) rect(\frac{\tau - \frac{2r_i(t_n)}{c}}{T_p}) \cdot \exp(j\pi k_r(\tau - \frac{2r_i(t_n)}{c})^2) rect(\frac{t_n}{T_a}) \exp(-j\frac{4\pi r_i(t_n)}{\lambda}) \quad (4)$$

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

In practical engineering calculation, the improved algorithm (Huang et al., 2004) of time-domain pulse coherent method is often applied; the principles as shown in Equation 5:

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

with

$$\begin{aligned} s_a(t_n,\tau) &= \sum_{i=1}^{M}\sigma_{0i}W_a(\theta_i) rect(\frac{t_n}{T_a}) \\ &\cdot \exp(-j(\frac{4\pi r_i(t_n)}{\lambda})) \cdot \delta(\tau - \frac{2r_i(t_n)}{c}) \\ s_r(\tau) &= rect(\frac{\tau}{T_p}) \exp(j\pi k_r\tau^2) \end{aligned} \quad (5)$$

where $f(.)$ is the Fourier transform operator, $f^{-1}(.)$ is inverse Fourier transform operator, and $S_r(\xi)$ 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.

In accordance with Equation 5, 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 is achieved by the inverse Fourier transform of results.
5. For all the azimuth time samples, equations (1) to (4) is repeated to get the whole simulated raw data.

Parallelizability analysis of SAR raw data simulation

According to stop-and-go model, SAR 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 MPI, grid computing or other parallel technologies.

The parallelism of SAR raw data simulation can be divided into coarse-grained and fine-grained strategy. The traditional parallel approach is a coarse-grained parallel approach, which takes repetitious transmitting and receiving pulses process as task. The method completes the task assignment through dispatching a reasonable simulated pulse number to different nodes, CPU, CPU core, as shown in Equation 6, in which D'_k represents the calculation task of node k. The parallel simulation with CUDA is a fine-grained parallel method, which optimizes the largest time-consuming step respectively. The task of every thread is the signal azimuth calculations of single scattering point and a single sampling point multiplication, as shown in Equation 7, where $D''_{(n,i)}$ is the azimuth signal of point i in time t_n, $D'''_{(n,j)}$ is the spectrum product of linear FM signal and azimuth signal at range gate j in time t_n.

$$s_r(t,\tau) = \sum_{k=0}^{m-1} D'_k = \sum_{k=0}^{m-1}\sum_{n=T_k}^{T_{k+1}} s_r(t_n,\tau) \quad (6)$$

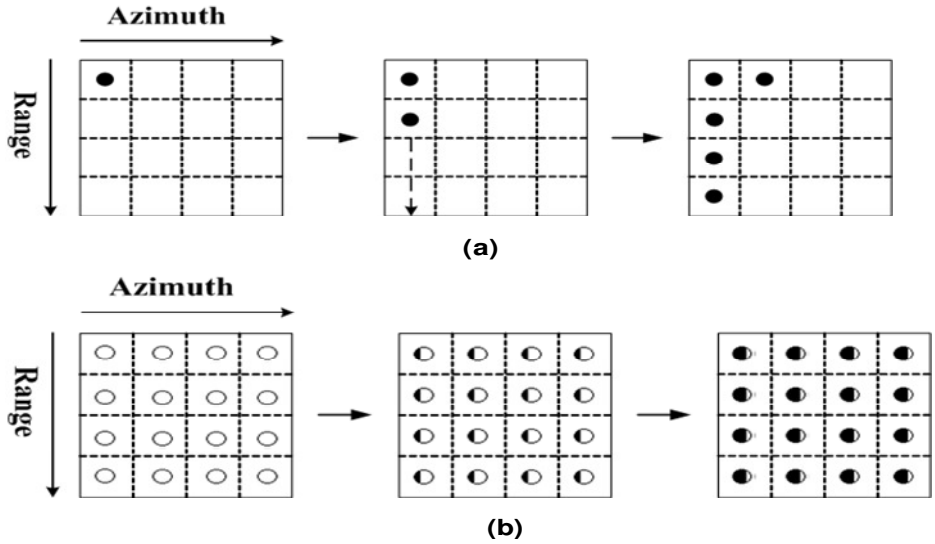


Figure 7. The serial simulation in CPU (a) and parallel simulation in GPU (b).

$$s_r(t,\tau) = \sum_{n=0}^{T} f^{-1}\{f[\sum_{i=1}^{M} D''_{(n,i)}] \square S(\xi)\} = \sum_{n=0}^{T} f^{-1}\{\sum_{j=0}^{N} D'''_{(n,j)}\} \quad (7)$$

According to Equation 7, a large number of GPU threads can be used for parallel execution in the four calculations to finish fine-grained subtask as follows:

1. Loop calculation of $D''_{(n,i)}$;

2. Fast Fourier transform of $\sum_{i=1}^{M} D''_{(n,i)}$, $s_r(t)$;

3. Complex vector multiplication for $\sum_{j=0}^{N} D'''_{(n,j)}$;

4. Inverse fast Fourier transform of $\sum_{j=0}^{N} D'''_{(n,j)}$.

As is shown in Figure 8, the GPU parallel computing from the aforementioned four steps and SAR raw data in one azimuth position are completed, thus, the loop, azimuth time and whole SAR raw data are simulated. Therefore, the main content of this GPU parallel method includes fine-grained tasks design and subtasks parallel speedup using CUDA language.

The SAR raw data fine-grained parallel simulation with CUDA not only accords with physical process of echo generating, but also takes full advantage of GPU hardware resources and computing power. The method with CUDA is well suited for large scene and complex scene airborne/spaceborne SAR raw data simulation.

Scalable parallel simulation with CUDA on GPU

As is shown in Figure 7, the conventional SAR raw data serial simulation calculates each scattering point echo contributing one-by-one in some azimuth moment, and then repeats at the next moment until the end of SAR flight. The CUDA-based SAR 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 improves the computing speed about 1 to 2 orders of magnitude finally.

To make use of obvious advantage of GPU parallel computing, the SAR raw data parallel simulation algorithm with CUDA is proposed, and applied to the time-consuming SAR complex scenes raw data simulation. The algorithm flow is as follows:

1. 3D target is placed in a flat rectangle to build scene geometry model firstly. Secondly geometric characteristics map is calculated using GPU automatic blanking and graphic transformation. Thirdly the scattering coefficient map of 3D target is calculated using GRECO algorithm. The scattering coefficient map of flat rectangle is represented by SAR image. Finally the two map are combined to be complex scene scattering coefficient map;
2. According to the system parameter, the linear FM signal is calculated, converted to frequency domain, and

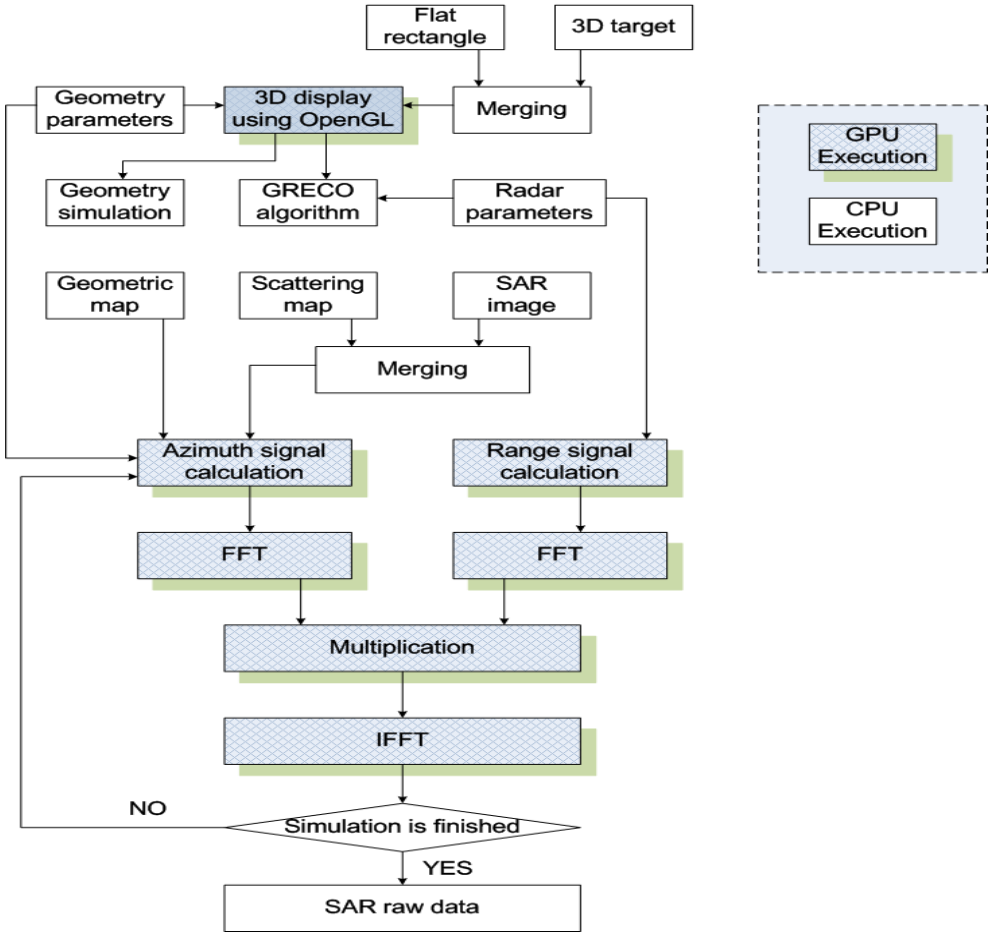


Figure 8. The algorithm diagram of scalable parallel SAR raw data simulation with CUDA.

transmitted to the GPU memory;
3. 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;
4. The azimuth signal spectrum is calculated by CUDA's 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, Equations (1) to (4) are repeated until the end of simulation time. The algorithm diagram is shown in Figure 8. In the figure, two types of markers indicate the simulation execution in GPU or CPU.

SIMULATION RESULTS

Through the introduction of CUDA, the technology can be introduced into SAR raw data simulation algorithm for large-scale parallel computation. The following three categories of airborne SAR raw data simulation experiments are designed to analyze the speedup, accuracy 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 1.

Parallel computation speed analysis

In order to test the speed of scalable parallel computing, the two-dimension flat surface target raw data simulation on GPU and CPU are executed respectively. There are ten groups of experiments in different 2D flat surface target scattering point matrix size, such as 1×1, 2×2, 4×4, 8×8, 16×16, 32×32, 64×64, 128×128, 256×256 and 512×512. The simulation results are shown in Figure 9. In Figure 9a, for small scattering point matrix, the computing time of CPU is slightly larger than GPU, but for large scattering point matrix, it is much larger than GPU. In Figure 9b, it can be seen that GPU acceleration increases

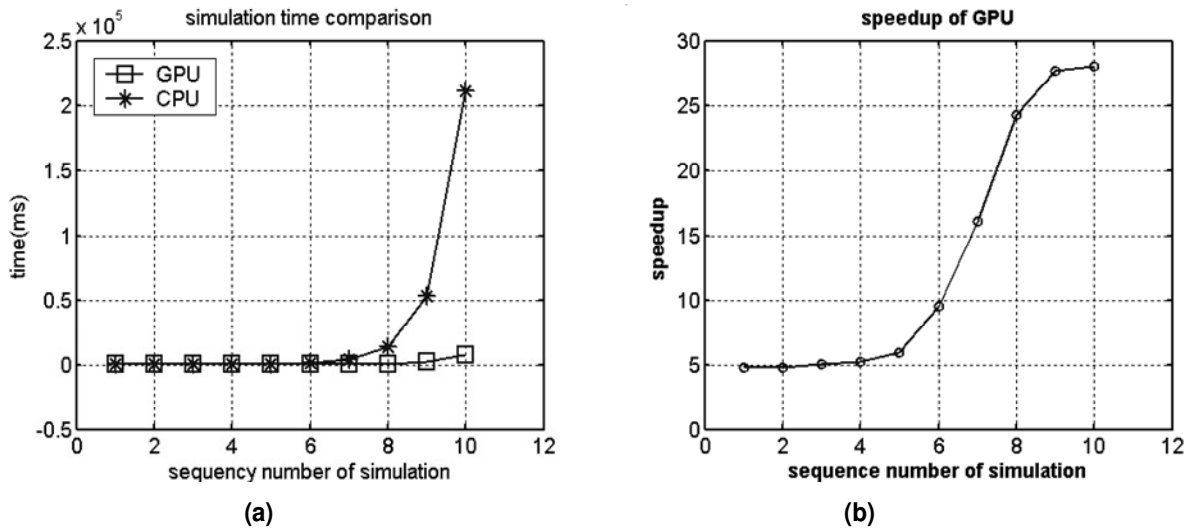


Figure 9. Time-consuming between GPU and CPU (a) and accelerated times of GPU to CPU (b).

Table 1. Simulation parameters.

Parameter	Value
Wave length	0.05 m
PRF	300 Hz
Pulse width	10 us
Band width	30 MHz
Sampling rate	60 MHz
Flight velocity	130 m/s
Center distance	10000 m
Scene size	1024×1024

with the amount of scattering points, and begins to decline when the number of scattering points reaches a certain degree. The first increased and then decreased of the accelerated growth rate is limited by the calculating ability of GPU hardware. GPU hardware has the limited parallel processing units. When there are fewer scattering points, GPU's calculating units have not been fully used. When there are lots of scattering points, GPU's calculating units cannot meet the simulation tasks. Thus part of the parallel task is waiting for calculation. That makes the decline in speed acceleration.

The three-dimension targets simulation is needed in the complex scene simulation and interferometric SAR simulation applications. So, a simple analysis of speed in three-dimension cone simulations will be given here. 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 100×100. The speed experiment results are shown in Table 2.

As can be seen in Table 2, speed acceleration in three-dimensional simulation is more obvious. 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.

It can be seen from the aforementioned that, the speedup of CUDA-based simulation to CPU target has reached 28.05 in two-dimension and 29.32 in three-dimension. To achieve the same acceleration, a cluster of equivalent 32 single-core CPU nodes should be constructed if you use the OpenMP, MPI, grid computing method. Furthermore, the CPU parallel method will cost 20 to 30 times more in hardware, electric power and space. So the SAR raw data scalable parallel simulation with CUDA is undoubtedly a high-performance computing solution which is space saving, energy-saving and low-cost.

Accuracy analysis

As the GPU and CPU hardware structure is different, the performance of floating-point operations is slightly different between them. Here, GPU parallel computing accuracy will be verified with the designed experiment. The experiment content includes the point target raw data simulation in CPU and GPU, imaging and assessment. The resolution, expansion coefficient, peak sidelobe ratio (PSLR) and integral sidelobe ratio (ISLR) are used for testing image quality, which is the indicator of GPU parallel computing accuracy.

The point target imaging quality results of CPU and

Table 2. Simulation speed comparison.

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

Table 3. Simulation accuracy comparison in azimuth direction.

Parameter	Res	Coefficient	PSLR	ISLR
CPU	1.057	0.883	-13.382	-9.990
GPU	1.058	0.883	-13.420	-10.059
Error	0.001	0.000	0.038	0.069

Table 4. Simulation accuracy comparison in range direction.

Parameter	Res	Coefficient	PSLR	ISLR
CPU	2.219	0.888	-13.290	-10.015
GPU	2.219	0.888	-13.301	-10.008
Error	0.000	0.000	0.011	0.007

GPU is shown in Tables 3 and 4, where 'Res' indicates the resolution, 'Coef' indicates the expansion coefficient. The resolution and expansion coefficient that is calculated by CPU and GPU is almost the same. There is a difference of 0.001 in azimuth resolution. Both the PSLR and ISLR of the two methods are just different from the second decimal place, in which the range direction has less errors and the azimuth direction has more errors. Summing up the aforementioned, the less errors of CPU and GPU simulation are existent, but are within the acceptable range. So the scalable parallel computation method with CUDA can meet the accuracy of SAR raw data simulation, and can be applied in research and engineering project.

Complex scene simulation

According to the method, the complex scene that is combined with a three-dimension object and natural scene will be simulated to verify the imaging results of simulated echo data. The experiment takes a civilian aircraft as three-dimensional object, and takes a MSTAR SAR image as the scattering coefficient map of natural scene. Through the SAR raw data parallel simulation of complex scene with CUDA and imaging, the final simulation results is shown in Figure 10.

Conclusion

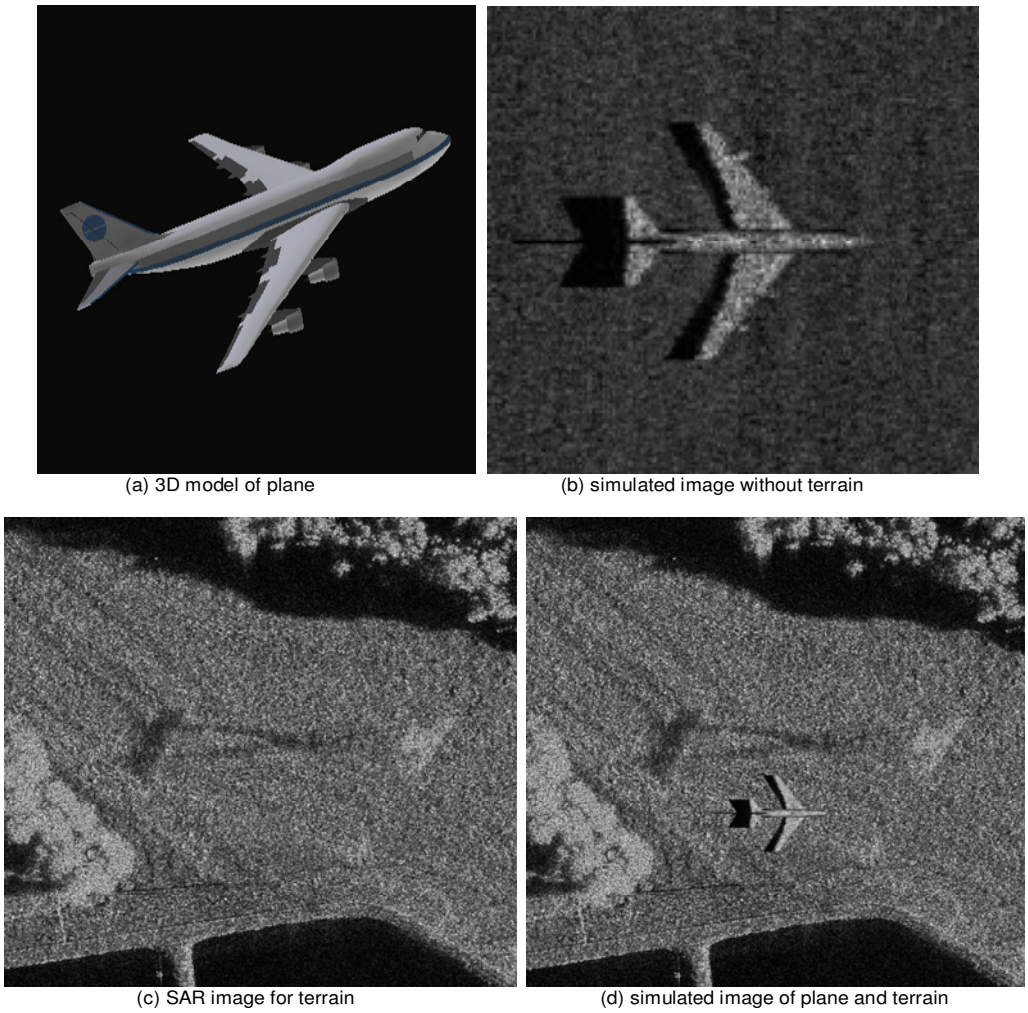
The GPU general purpose computing has been more widely used with the development of CUDA architecture. The paper introduces hybrid CUDA technology and computer graphics technology to solve calculation bottle-neck and simulation accuracy of SAR complex scene raw data simulation.

The method exploits the general purpose computing and graphics purpose capability of GPU, and to solve the geometry modeling, scattering calculation, raw raw data calculation in SAR simulation. The experiments results show that the method is 4 to 29 times faster than CPU methods under high precision.

Furthermore, the method only needs to upgrade the computer with one GPU card, which has 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 SAR/InSAR raw data simulation of real terrain, which assists the spaceborne SAR/InSAR system design and validation.

ACKNOWLEDGMENTS

This work is supported in part by the National High-tech R&D Program of China (863 Program) under grant No.2007AA120302 and the National Basic Research Program of China (973 Program) under grant No. 2009CB724003.



(a) 3D model of plane (b) simulated image without terrain

(c) SAR image for terrain (d) simulated image of plane and terrain

Figure 10. Imaging results of SAR complex scene raw data simulation based on GPGPU and CG.

REFERENCES

Franceschetti G, Migliaccio M, Riccio D, Gilda S (1992). SARAS: a synthetic aperture radar (SAR) raw signal simulator. IEEE Trans.On Geosci. Remote Sens., 30(1): 110-123.

Garland M, Le Grand S, Nickolls J, Anderson J, Hardwick J, Morton S, Phillips E, Yao Z, Volkov V (2008). Parallel computing experiences with CUDA. IEEE Micro. 28(4): 13-27.

Gray MA (2009). Getting start with GPU programming. Comput. Sci. Eng., 11(4): 61-64.

Huang LS, Wang ZS, Zheng TY (2004). A fast algorithm based on FFT used in simulation of SAR return wave signal. J. Remote Sens., 18(2): 128-136.

Liu YT (1999). Radar imaging technology. Harbin Institute of Technology Press, Harbin. pp. 248-256.

Margarit G, Mallorqui JJ, Lopez-Martinez C (2007). GRECOSAR, a SAR simulator for complex targets: application to urban environments. IGARSS2007. pp. 4160-4163.

Margarit G, Mallorqui JJ, Rius JM, Jesus SM (2006). On the usage of GRECOSAR, an orbital polarimetric SAR simulator of complex targets, to vessel classification studies. IEEE Trans. Geosci. Remote Sens., 44(12): 3517-3526.

Rius JM, Ferrando M, Jofre L (1993). GRECO: Graphical electromagnetic computing for RCS prediction in real time. IEEE Antennas Propag. Mag., 35(2): 7-17.

Su Y, Qi XY (2008). OpenMP based space-borne SAR raw signal parallel simulation. J. Graduate School of the Chinese Academy of Sci., 25(1): 129-135.

Timo B, Uwe S (2009). Hybrid GPU-Based single- and double-bounce SAR simulation. IEEE Trans. On Geoscience and Remote Sensing. 47(10): 3519-3529.

Wang XS, Huang LS, Wang ZS (2006). Research on parallel arithmetic of distribute space borne SAR ground target simulation. J. Syst. Simul., 18(8): 2097-2100.

Wu C, Liu KY, Jin M (1982). Modeling and a correlation algorithm for spaceborne SAR signals. IEEE Trans. on Aerospace and Electronic Systems. 18(5): 563-574.

Zhang F, Bai L, Hong W (2008a). InSAR imaging geometry simulation based on computer graphics. ISPRS 2008. pp. 781-784.

Zhang F, Hong W, Li DJ (2008b). SAR image simulation of man-made scenes based on computer graphics. IGARSS 2008. IV: 1395-1397.

Zhang F, Lin Y, Hong W (2008c). SAR echo distributed simulation based on grid computing. J. Syst. Simul., 20(12): 3165-3170.

Zhang F, Wang BN, Xiang MS (2010). Accelerating InSAR raw data simulation on GPU using CUDA. IGARSS 2010, pp. 2932-2935.