GPU based FDTD method for investigation on the electromagnetic scattering from 1-D rough soil surface

C.-G. Jia, L.-X. Guo, J. Li
School of Science, Xidian University
No. 2, Taibai Road
Xi'an, Shaanxi 710071 China

Abstract- In this paper, the graphic processor unit (GPU) implementation of the finite-difference time domain (FDTD) algorithm is presented to investigate the electromagnetic (EM) scattering from one dimensional (1-D) Gaussian rough soil surface. The FDTD lattices are truncated by uniaxial perfectly matched layer (UPML), in which the finite-difference equations are carried out for the total computation domain. Using Compute Unified Device Architecture (CUDA) technology, significant speedup ratios are achieved for different incident frequencies, which demonstrates the efficiency of GPU accelerated the FDTD method. The validation of our method is verified by comparing the numerical results with these obtained by CPU, which shows favorable agreements.

I. INTRODUCTION

Nowadays, the investigation on the statistical characteristic of electromagnetic scattering from random rough surface has attached considerable interest owing to its significant applications in the fields of remote sensing, target identification and radar detection [1]. Many kinds of methods, including analytical and numerical approaches, have been carried out to deal with the electromagnetic scattering model. Taking the approximate analytical methods for example, the Kirchhoff approximation [2], the small-slope approximation (SSA) [3], and the small-perturbation method (SPM) [4] have been studied, but they are usually limited by roughness, incident angle, and low precision. Numerical methods are widely employed to calculate the model, such as the parallel method of moment (MoM) as well as its accelerated method[5], the Generalized Forward-Backward Method (GFBM) , and finite-difference time domain (FDTD) method [6].

Compared with other numerical methods, the FDTD method has its own advantages in analyzing the scattering from rough surface [6]. The traditional FDTD method is too week to deal with the electrically large problem due to limitation of computation time. The MPI-based parallel FDTD was presented by J. Li *et al*. [6]. Using the method, the computation time is extremely reduced compared to sequential implementation. However, the speedup ratio of MPI-based method is limited by numbers of Central Processing Unit (CPU) processes. Fortunately, Compute Unified Device Architecture (CUDA) technology based on GPU has been extensively implemented for the large scale FDTD simulations successfully [7]. Compared to the MPI technology, graphic processor unit (GPU) can achieve huge speedup ratios at low cost for its powerful computing capability, which motives us into adopt the GPU-based FDTD technology for analyzing the scattering from rough surface. Up to now, to our knowledge, few works have been reported to solve this problem using the GPU-based FDTD implementation. Additionally, a uniaxial perfectly matched layer (UPML) medium is employed to truncate the FDTD lattices, and the finite-difference equations in the UPML medium are used for the total computation domain making the parallel algorithm convenient to implement. In this paper, the precision of calculations performed on both GPU and CPU is single precision arithmetic.

II. THEORETICAL ANALYSIS

A. *Gaussian rough surface model*

In order to investigate the characteristic of EM scattering from rough surface, the first step is to generate the profile of 1-D rough surface, which is simulated by the Monte Carlo method. Taking the TM incidence for example, the scattering model for one dimensional (1-D) random rough surface with height profile function $y = f(x)$ is shown in Fig. 1, where $f(x)$ is a Gaussian distributed rough surface with the Gaussian power spectrum density function $W(K)$ expressed as follows [8]

$$W(K) = \frac{\sigma^2 l}{2\sqrt{\pi}} \exp\left(-\frac{K^2 l^2}{4}\right) \tag{1}$$

Where σ and l are root mean square (rms) height and correlation length, respectively, by which the profile of rough surface is determined. L is the length of rough surface we simulate. In order to model the scattering from an infinite surface, Gaussian window function is introduced and expressed as [9]

$$G(x,y) = \exp\left\{-\left[\left(x-x_{cen}\right)^2 + \left(y-y_{cen}\right)^2\right]\left(\frac{\cos\theta_i}{T}\right)^2\right\} \tag{2}$$

Where x_{cen} and y_{cen} are the center coordinates of the connective boundary. T is a constant which determines the tapering width of the window function so chosen that the tapering drops from unity to 10^{-3} at the edge, as well as $\cos\theta_i/T=2.6/\rho_m$, where ρ_m is the minimum distance from the center coordinate to the edge of the connective boundary .

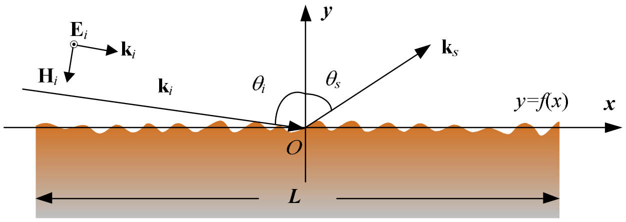

Fig. 1. Geometry for EM scattering from 1-D rough surface (TM wave).

B. *FDTD method for rough surface*

Fig. 2 depicts the division model of computation region for the FDTD algorithm in calculating electromagnetic scattering from rough surface. In order to simulate the infinite free space in the finite computing field, a virtual absorbing boundary is employed outside the FDTD region. There have been many absorbing boundary conditions adopted in the FDTD method. In this paper, the UPML absorbing medium are employed to truncate the FDTD lattices. The connective boundary is needed to divide the computation region into the total field region and the scattered field region [10], where the incident wave is generated. After the near fields are obtained, far fields can be achieved by doing a near-to-far field transformation at the output boundary [10]. Finally, the bistatic scattering coefficient in the far zone is calculated by [6].

$$\sigma = \lim_{r\to\infty} \frac{2\pi r}{L} \frac{|E_s|^2}{|E_i|^2} \tag{3}$$

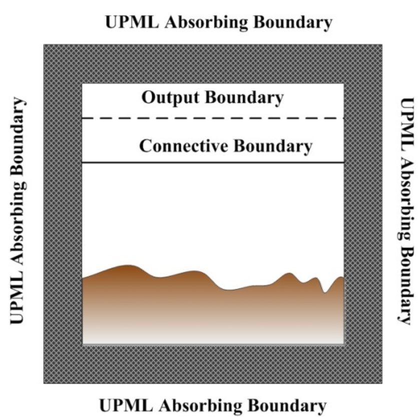

Fig. 2. FDTD model of 1-D rough surface

III. GPU IMPLEMENTATION OF FDTD ALGORITHM

In this section, the parallel computing platform and programming model CUDA is introduced. The parallelization strategy of the GPU based FDTD method for EM scattering from rough surface is clarified in detail, which includes the management of device memory such as the global, shared and texture memory, as well as the kernels implementation.

A. *CUDA programming model*

The introduction of NVIDIA' GPU based on CUDA architecture gave rise to a new era of graphics computing, without esoteric knowledge of graphics computation models. CUDA is a highly parallel and efficient computing architecture, with which GPUs can solve many complex problems by built-in streaming multiprocessors executing a number of threads in parallel [11]. The CUDA programing model assumes that the sequential code executes on the host (CPU) while the instruction with high data parallelism executes on the device (CUDA-enabled GPU).

As illustrated by Fig. 3, a CUDA program begins with serial execution on the host, including CPU and GPU memory allocation, initialization as well as de-allocation. Kernels defined as functions are executed on the device by a large amount of threads in parallel. The memories on the two platforms (host and device) are separated physically in the CUDA programming model. For further instructions on the CUDA technology, one can refer to [11].

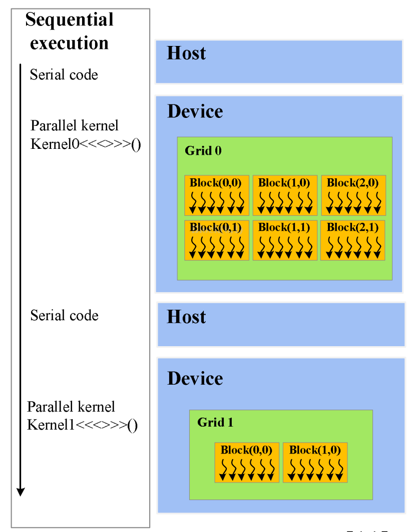

Fig. 3. The CUDA programing model [11]

B. *Memory management*

From the point of view of performance optimization for the FDTD approach, it is of great significance to optimize the usage of device memory to obtain best performance on GPU. CUDA device memory consists of various memory spaces, including global, local, constant, shared, texture, and registers memory, which have their own characteristic. In this paper, global, shared, and texture memory is utilized to achieve high GPU performance. Global memory can be written and read by the host by the application programming interface (API) functions. The transactions of global memory access should be coalesced and bandwidth is low due to residing in uncached off-chip memory. In order to boost the performance of the kernels, the on-ship shared memory is utilized to eliminate the uncoalesced access. Shared memory is available to the thread block, in which threads share their results and the execution of threads in the threadblock can be synchronized in the block level. Taking TM^z case for example, as demonstrated in Fig.

4, data is first loaded from global memory to shared memory when the E field and the H field update is executed. When the H components (H_x, H_y) calculated, the E_z values of the current block of threads are not only copied to shared memory, values of the left column threads of the right adjacent block and the up row thread of the down adjacent block are also loaded. When the E field iteration function is invoked, the H_x and H_y values of the current block are not only transferred from global memory to shared memory, but H_x values of the down row threads of the up adjacent block and H_y values of right column of the left adjacent block are also delivered. Finally, further improvement of the speedup ratio is achieved by utilizing texture memory, which is read-only cache

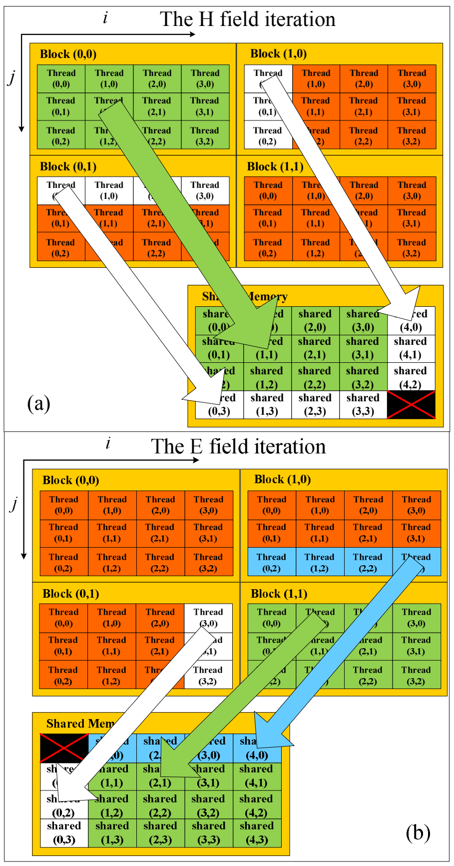

Fig. 4. Data transfers from global memory to shared memory

C. Device kernels

First, as illustrated by Fig. 5, Gaussian rough model is built by the Monte Carlo method presented in Section 2.1. And then, CPU assigns the host and device memory, as well as grid and block size based on the model. Parallel implementation is carried out when referring to the near field iteration, which is extremely time-consuming in the whole FDTD computation. It is necessary to synchronize for some threads to share data with each other. The threads in the same block synchronize by using *__syncthreads ()* though shared memory, while a new kernel function is invoked to synchronize though global memory for the threads belonging to different blocks. In order to force synchronization on the grid level, five *kernels* are utilized to achieve the functions, including *IncidentHKernel* (the incident magnetic field update), *IncidentEKernel* (the incident electric field update), *ConnectionKernel* (introducing the incident wave at the connective boundary), *eKernel* (the electric field component(s) update), and *hKernel* (the magnetic field component(s) update). When the near field iteration is finished by GPU, the far field can be obtained with great ease on the CPU platform

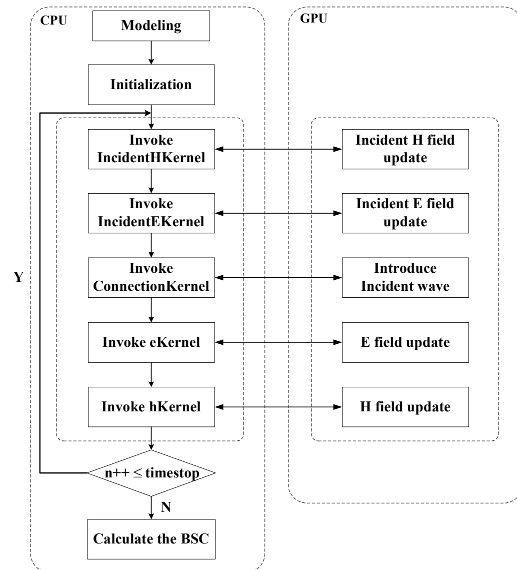

Fig. 5. The flowchart of GPU based FDTD algorithm for rough surface

IV. NUMERICAL RESULTS AND DISCUSSION

. In this section, the numerical results of EM scattering from 1-D soil surface by GPU based FDTD method are discussed in detail. To ensure the accuracy and stability of the FDTD method, the spatial increment and time increment are taken as $\Delta x = \Delta y = \Delta = \lambda / 20$ and $\Delta t = 0.5 \times \Delta / c$, respectively. λ is the incident wavelength and c is the light speed in vacuum. The UPML thickness is 8Δ.

The accuracy of the CUDA implementation of FDTD algorithm is illustrated by comparing the numerical results with those obtained by sequential execution on CPU. Fig. 6 demonstrates the bistatic scattering from a Gaussian soil surface with characteristic parameters $\delta = 0.1\lambda$ and $l = 1.0\lambda$ under the incident angle $\theta_i = 35^\circ$ at the incident frequency of $f = 8\text{GHz}$. The generated length of rough surface is $L = 1638.4\lambda$ (32768Δ). The relative permittivity of surface corresponding to the soil with 3.8% moisture is taken as $\varepsilon_r = (2.5, 0.18)$. The results averaged by 20 surface realizations are in good agreement by the two implementations for both *TM* and *TE* incidence, demonstrating the accuracy of our FDTD-CUDA implementation. The time consuming of traditional FDTD scheme are approximately 112.91 and 119.67 hours for *TM* and *TE* case respectively. By contrast, the computation time of GPU based FDTD are 1.51 and 1.54 hours for the two incident cases. Additional, we can easily find that the time cost reduces dramatically by the GPU implementation.

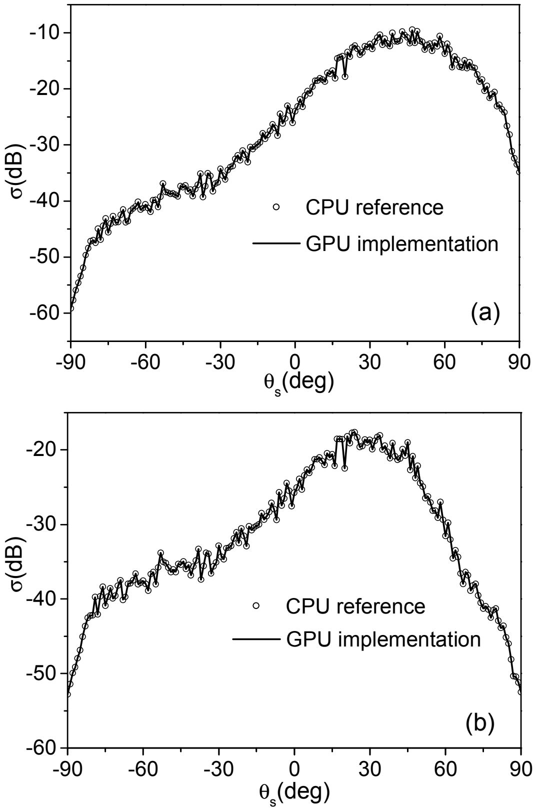

Fig. 6. Comparisons of the bistatic scattering from a soil surface by two implementations: (a) *TM* case; (b) *TE* case.

Furthermore, the efficiency of our implementation is demonstrated by comparison of the CPU and GPU time for calculating the EM scattering from rough surface as incident frequency increases from 1GHz to 16GHz under the incident angle of θ_i=55°. And the mash along x direction increases from 4096 to 65536 by keeping the length of rough surface $L = 61.44\text{m}$. The C program is executed on Intel Core2 Quad Q8300, 2.5GHz and 3.5GB of RAM. The kernels are performed on a NVIDIA GeForce GTX570 GPU card, which has 480 stream processors and 1280 MB video memory. Table I compares computation time of serial FDTD method for rough surface with one surface realization with that of GPU implementation. As illustrated by the table, it is obviously observed that speedup factor increases with the increase of unknowns, which demonstrates that huge computations can make full use of thousands of threads on the GPU. In addition, the GPU accelerated FDTD implementation has also striking advantage versus the message-passing-interface (MPI)-based FDTD scheme [6] that the speedup factor is in direct proportion to the number of CPU process used for computation, especially when solving the problem of electrically large rough surface.

V. CONCLUSIONS

In this paper, GPU accelerated FDTD method is extended to investigate the electromagnetic scattering from 1-D rough soil surface. Global, shared, and texture memory is utilized to optimize the performance of GPU. The accuracy of the method is demonstrated by comparing the result obtained by sequential execution on the CPU platform. Compared to the previous work, MPI-based FDTD method, our implementation can achieve favorable speedup factor. In our future work, the investigation will be focused on the electromagnetic scattering from 2-D randomly rough surface, which has more practical applications.

Table I
Comparison of CPU and GPU time (TM case).

Incident frequency (GHz)	Mesh along x	Mesh along y	CPU time(s)	GPU time(s)	speedup
1	4096	100	265.94	7.82	34.00
4	16384	100	3960.13	63.45	62.41
8	32768	100	15726.96	214.97	73.15
16	65536	100	62961.59	790.25	79.67

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No.61225002), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No.20100203110016), and the Fundamental Research Funds for the Central Universities (Grant No.K50510070001).

REFERENCES

[1] Ku, H.C., Awadalah, R.S., Mcdonald, R.L., and Woods, N.E., "Fast and accurate algorithm for electromagnetic scattering from 1-D dielectric ocean surface," *IEEE Trans. Antennas Propag.*, vol. 54, pp. 2381-2391, 2006.

[2] A. K. Sultan-Salem and G. L. Tyler, "Validity of the Kirchhoff approximation for electromagnetic wave scattering from fractal surfaces," *IEEE Trans. Geosic. Remote Sens.*, vol. 42, pp. 1860-1870, 2004.

[3] Y. Q. Wang and S. L. Broschat, "A systematic study of the lowest order small slope approximation for a Pierson-Moskowitz spectrum," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, pp. 158-162, 2011.

[4] L. X. Guo, Y. Liang, J. Li, and Z. S. Wu, "A high order integral SPM for the conducting rough surface scattering with the tapered wave incidence TE case," *Progress In Electromagnetics Research, PIER*, vol. 114, pp. 333-352, 2011.

[5] L. X. Guo, A. Q. Wang, and J. Ma, "Study on EM scattering from 2-D target above 1-D large scale rough surface with low grazing incidence by parallel MOM based on PC clusters," *Progress In Electromagnetics Research, PIER*, vol. 89, pp. 149-166, 2009.

[6] J. Li, L. X. Guo, H. Zeng, and X. B. Han, "Message-passing-interface-based parallel FDTD investigation on the EM scattering from a 1-D rough sea surface using uniaxial perfectly matched layer absorbing boundary," *J. Opt. Soc. Am. A*, vol. 23, pp. 359-369, 2009.

[7] Piotr Sypek, Adam Dziekonski, and Michal Mrozowski, "How to render FDTD computations more effective using a graphics accelerator," *IEEE Trans. Magn.*, vol. 45, pp. 1324-1327, 2009.

[8] Thorsos, E.I., "The validity of the Kirchhoff approximation for rough surface scattering using a Gaussian roughness spectrum," *J. Acoust. Soc. Am.*, vol. 83, pp. 78-92, 1988.

[9] Adrian K. Fung, Milind R. Shah, Saibun Tjuatia, "Numerical simulation of scattering from three-dimensional random rough surface," *IEEE Trans. Geosic. Remote Sens.*, vol. 32, pp. 986-994, 1994.

[10] Taflove, A. and S. C. Hagness, *Computational Electrodynamics: The Finite-difference Time-domain Method*, Artech House, Boston, 2005.

[11] "*NVIDIA CUDA C Programming Guide*," ver. 4.2, NVIDIA Corporation, May. 2012.