

Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

**Graphics processor unit accelerated
finite-difference time domain method
for electromagnetic scattering from
one-dimensional large scale rough
soil surface at low grazing incidence**

Chungang Jia
Lixin Guo
Ke Li

Graphics processor unit accelerated finite-difference time domain method for electromagnetic scattering from one-dimensional large scale rough soil surface at low grazing incidence

Chungang Jia,* Lixin Guo, and Ke Li

Xidian University, School of Physics and Optoelectronic Engineering, Department of Radio Physics, South 2 Taibai Road, Xi'an 710071, China

Abstract. The graphics processor unit-based finite-difference time domain (FDTD) algorithm is applied to study the electromagnetic (EM) scattering from one-dimensional (1-D) large scale rough soil surface at a low grazing incident angle. The FDTD lattices are truncated by a uniaxial perfectly matched layer, and finite difference equations are employed in the whole computation domain for convenient parallelization. Using Compute Unified Device Architecture technology, we achieve significant speedup factors. Also, shared memory and asynchronous transfer are used to further improve the speedup factors. Our method is validated by comparing the numerical results with those obtained by using a CPU. The influences of the incident angle, correlation length l , and root-mean-square height δ on the bistatic scattering coefficient of a 1-D large scale rough surface at low grazing incidence are also discussed. © 2014 Society of Photo-Optical Instrumentation Engineers (SPIE) [DOI: [10.1117/1.JRS.8.084795](https://doi.org/10.1117/1.JRS.8.084795)]

Keywords: finite-difference time domain; graphics processor unit; low grazing incidence; soil surface.

Paper 14374SS received Jun. 26, 2014; revised manuscript received Sep. 2, 2014; accepted for publication Oct. 7, 2014; published online Nov. 5, 2014.

1 Introduction

Investigations on electromagnetic (EM) randomly rough surface have become a popular topic owing to significant applications in the fields of remote sensing, target identification, and radar detection.¹⁻³ Many analytical and numerical approaches have been developed to deal with the EM scattering model. For example, the Kirchhoff approximation,^{4,5} which is valid when a rough surface is smooth, and the small-perturbation method⁶ used where the standard deviation of a rough surface is small compared with the wavelength, are invalid at low grazing incident angles. To solve this scattering problem, numerical methods, such as the parallel method of moment (MoM) based on the message passing interface (MPI) between the personal computer (PC) clusters,⁷ the generalized forward-backward method,⁸ the multilevel sparse-matrix canonical-grid method,⁹ and the MPI-based parallel finite-difference time domain (FDTD) method¹⁰ are extensively used.

This paper presents the graphics processor unit (GPU)-accelerated parallel FDTD method to study the scattering characteristic of the bistatic scattering coefficient. The proposed approach differs from the previously mentioned methods in that it studies the bistatic one-dimensional (1-D) large scale rough surface based on the GPU platform using Compute Unified Device Architecture (CUDA) technology.

Compared with others, the FDTD method has its own advantages.¹⁰ When a large scale rough surface with low grazing incident angles is investigated, the generated length of rough surface should be as long as possible,¹¹ which results in large numbers of unknowns. The traditional FDTD method can hardly handle such problems because of the limitation of computation time. Using the MPI-based parallel FDTD mentioned above,¹⁰ the computation time is extremely

*Address all correspondence to: Chungang Jia, E-mail: jcg.0502@163.com

reduced compared to that of sequential implementation. However, the speedup factors of the MPI-based method are limited by the high cost of the hardware. Fortunately, CUDA technology based on GPU has been extensively and successfully implemented for large-scale FDTD simulations.¹²⁻¹⁴ Compared to the MPI technology, the GPU can achieve huge speedup factors at a low cost for its powerful computing capability, which is why we adopted GPU-based FDTD technology to extend the application of the FDTD method in analyzing scattering from a large scale rough surface at low grazing incident angles. To our knowledge, few studies have been reported to solve this problem using the GPU-based FDTD implementation. Here, a uniaxial perfectly matched layer (UPML) medium is used to truncate the FDTD lattices, and the finite difference equations in the UPML medium are used for the total computation domain to facilitate the implementation of the parallel algorithm. All of our calculations are single precision arithmetic.

The remainder of this paper is organized as follows: in Sec. 2, the theoretical equations for calculating EM scattering from rough surface by FDTD are presented in detail. In Sec. 3, a programmable GPU-based CUDA architecture is introduced, and details about the implementations of the GPU accelerated FDTD for a rough surface are illustrated. Shared memory and asynchronous transfer are used to improve the performance. Also, the influences from the incident angle, correlation length, as well as the root-mean-square (rms) height to the bistatic scattering coefficient are also discussed in Sec. 4. Some concluding remarks are addressed and further investigations are proposed in Sec. 5.

2 Theoretical Analysis

2.1 Rough Surface Model

We generate the profile of a 1-D rough surface, which is simulated by the Monte Carlo method. Taking the TM incident wave, for example, the scattering model for a 1-D random rough surface with a height profile function $y = f(x)$ is shown in Fig. 1, where an incident wave impinges on the surface in the direction of \mathbf{k}_i , which makes angle θ_i relevant to the y -axis. The scattered direction is \mathbf{k}_s and the scattered angle is θ_s . $f(x)$ is a Gaussian distributed rough surface with the exponential power spectrum density function $W(K)$ expressed as follows:

$$W(K) = \frac{\delta^2 l}{\pi(1 + K^2 l^2)}, \tag{1}$$

where the quantities δ and l are rms height and correlation length, respectively, and determine the profile of the rough surface. L is the length of the rough surface. As shown in Fig. 1, in order to avoid the edge diffraction effect, a Gaussian window function is introduced and expressed as¹⁵

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

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, and is chosen so 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.¹⁶

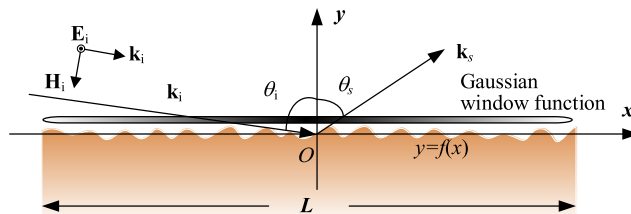


Fig. 1 Geometry for electromagnetic scattering from a one-dimensional (1-D) rough surface (TM wave).

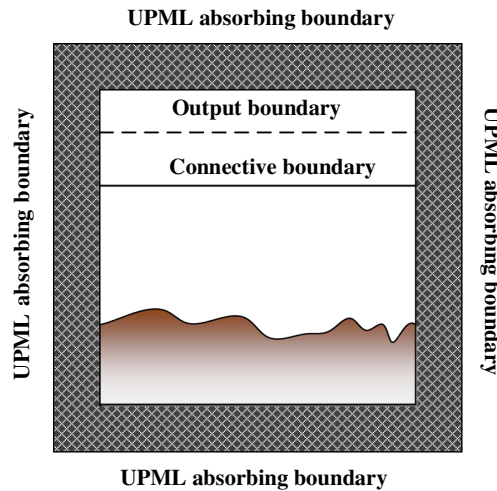


Fig. 2 Finite-difference time domain (FDTD) model of a 1-D rough surface.

2.2 FDTD Method for Rough Surface

Figure 2 shows the division model of the computation region for the FDTD algorithm used to calculate EM scattering from a rough surface. To simulate the infinite free space in the finite computing field, a virtual absorbing boundary is employed outside the FDTD region. We use the UPML absorbing medium^{17,18} to truncate the FDTD lattices. The connective boundary divides the computation into the total field region and the scattered field region,¹⁹ where the incident wave is generated. After the near fields are obtained, far fields can be determined by performing a near-to-far-field transformation at the output boundary.¹⁹ Finally, the bistatic scattering coefficient σ in the far zone is calculated by²⁰

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

where E_s is the scattered electric field and E_i is the incident electric field. r is the distance from the scatterer point to the origin.

3 CUDA Implementation of FDTD for Rough Surface

This section introduces the PC platform and CUDA programming model. The parallelization strategy includes CUDA implementation and computing optimization. Also, the performance is further improved by using shared memory and asynchronous transfer.

The introduction of the GPU-based CUDA architecture by NVIDIA 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 through built-in streaming multiprocessors executing a number of threads in parallel.²¹ The CUDA programming 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, and deallocation. 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 physically separated in the heterogeneous programming model. For further information about the CUDA technology, the reader can refer to Ref. 21.

As illustrated by Fig. 4, an exponential rough model is first built by the Monte Carlo method presented above. The CPU then assigns the host and device memory, as well as the 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. The near-field iteration includes the incident magnetic field update, the incident electric field update,

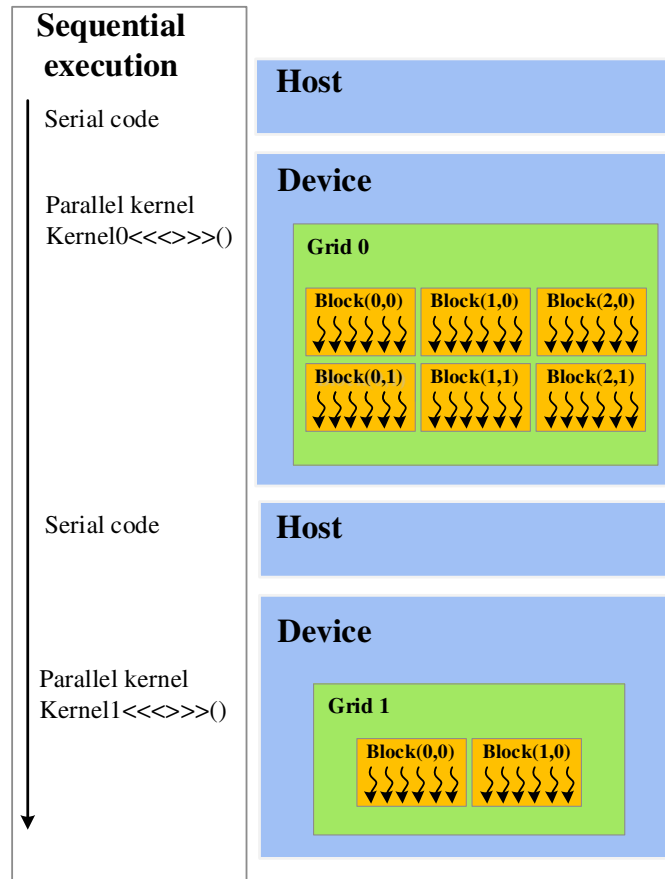


Fig. 3 Heterogeneous programming.

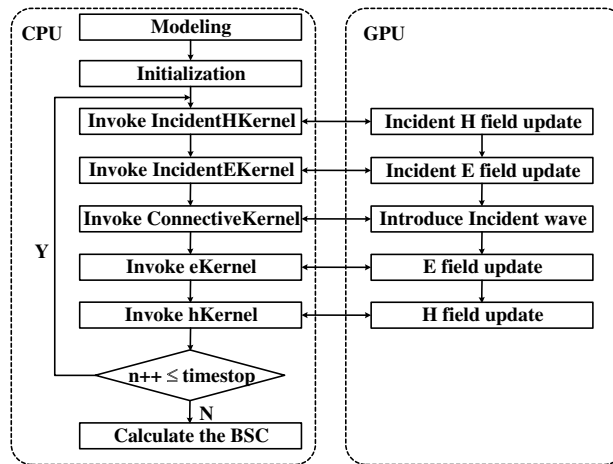


Fig. 4 The flowchart of the graphics processor unit-based FDTD algorithm for a rough surface.

introduction of the incident wave at the connective boundary, the electric field component(s) update, and the magnetic field component(s) update. 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, whereas a new kernel function is invoked to synchronize though global memory for the threads belonging to different blocks. 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

Table 1 Parameters of the computing platform.

<i>Host</i>	
CPU	Intel Xeon CPU E5-2620
Memory	32 Gb
<i>Device</i>	
GPU	NVIDIA Tesla k40c
Number of CUDA cores	2880
Total amount of global memory	11,520 Mbytes
Total amount of shared memory per MP	49,152 bytes
Total amount of registers available per MP	65,536

electric field component(s) update), and *hKernel* (the magnetic field component(s) update). When the near-field iteration is finished by the GPU, the far-field can be obtained with great ease on the CPU platform.

The CUDA implementation of FDTD for calculating EM scattering from the soil surface is performed on NVIDIA Tesla k40c with 2880 CUDA cores. Also, the sequential program is executed on Intel Xeon CPU E5-2620 2.10 GHz. The computing platform is listed in Table 1. The speedup factor in this paper is defined as the ratio of computation time for one surface sample by sequential FDTD to that by CUDA FDTD.

Taking the TM case, for example, the CPU and GPU times are compared for calculating the EM scattering from a rough surface as incident frequency increases from $f_i = 1$ GHz to $f_i = 64$ GHz at an incident angle of $\theta_i = 55$ deg. The mesh along the x -direction increases from 4096Δ to 262144Δ by keeping the length of the rough surface $L = 61.44$ m. Table 2 compares computation times of the serial FDTD method for a rough surface with one surface realization with that of the GPU implementation. As illustrated by the table, it is obvious that the speedup factors increase with an increase in the number of unknowns, but is reduced from 89.32 to 88.96 for 131,072 and 262,144 unknowns, which demonstrates that huge computations can make full use of the thousands of threads on the GPU and that the large data transfer between the host and the device reduces the speedup factors.

3.1 Further Improvement with Shared 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 the threads share their results and the execution of threads in the threadblock can be synchronized at the

Table 2 Comparison of CPU and GPU times with one surface realization.

f_i (GHz)	Mesh (Δ)	CPU time	GPU time	Speedup
1	4096	237.35	7.69	27.07x
4	16,384	3576.23	65.24	54.81x
8	32,768	14532.75	224.49	64.73x
16	65,536	59636.62	810.78	73.55x
32	131,072	241638.54	3243.53	74.49x
64	262,144	1039048.13	14524.21	71.53x

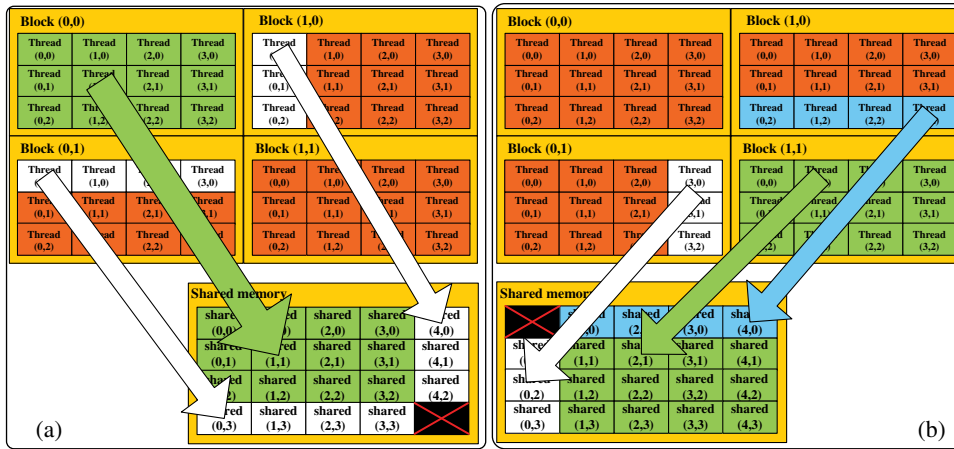


Fig. 5 Data transfers from global memory to shared memory: (a) magnetic field iteration and (b) electric field iteration.

block level. With the TM case as an example, Fig. 5 shows that the data are first loaded from global memory to shared memory when the electric field and magnetic field updates are executed. When the magnetic components (H_x, H_y) are calculated, not only are the E_z values of the current block of threads copied to the shared memory, but the 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 electric field iteration function is invoked, not only are the H_x and H_y values of the current block 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. The speedup factors as improved by shared memory are demonstrated in Table 3.

3.2 Further Improvement with Asynchronous Transfer

As shown in Tables 1 and 2, when the numbers of the meshes are 262144, the time for data transfer from CPU to GPU becomes prominent. Taking the TM case, for example, to achieve the far-field, the values of the E_z component and H_x component are needed to copy back from the GPU to CPU to perform a near-to-far field transformation. The asynchronous transfer is used to hide data transfers between the GPU and CPU by concurrently executing CUDA streams. Using multiple streams, the data transfer and computation can be overlapped. In this paper, the computation region is divided into n subgrids, and n is the number of streams. Figure 6 illustrates the C codes for a asynchronous transfer. It should be pointed out that the “offset_boundary” is the value of the last subgrid needed in the current subgrid update. The speedup factors as improved by asynchronous transfer are listed in Table 4.

Table 3 Speedup improvement with shared memory.

f_i (GHz)	Mesh (Δ)	CPU time	GPU time	Speedup
1	4096	237.35	7.69	30.86x
4	16,384	3576.23	56.25	63.57x
8	32,768	14532.75	193.53	75.09x
16	65,536	59636.62	704.95	84.59x
32	131,072	241638.54	2810.73	85.97x
64	262,144	1039048.13	12664.78	82.04x

```

for(itimestep=0; itimestep<ntimestep; itimestep++){
    .
    .
    .
    offset=0;
    for(i=0;i<nstreams;i++){
        offset=i*subthread*subgrids+offset_boundary;
        eKernel<<grids,threads,0,sream[i]>>>(d_uz+offset,d_pz+offset,d_p1z+offset,d_hx+
            offset,d_hy+ offset,d_hcb);

        cudaThreadSynchronize();
        hKernel<<grids,threads,0,sream[i]>>>(d_uz+offset,d_hx+offset,d_hy+offset,d_bx+
            offset,d_by+offset,d_ecb);

        cudaThreadSynchronize();
    }
}
for(i=0;i<nstreams;i++){
    offset=i*subthread*subgrids;
    cudaMemcpyAsync(h_uz+offset,d_uz+offset, size,cudaMemcpyDeviceToHost,stream[i]);
    cudaThreadSynchronize();
    cudaMemcpyAsync(h_hx+offset,d_hx+offset, size,cudaMemcpyDeviceToHost,stream[i]);
    cudaThreadSynchronize();
}
    
```

Fig. 6 C codes for realizing the asynchronous transfer.

4 Electromagnetic Scattering From Soil Surface at Low Grazing Incidence

To ensure the accuracy and stability of the FDTD method, the spatial and time increments are taken as $\Delta x = \Delta y = \Delta z = \lambda/20$ and $\Delta t = 0.5 \times \Delta z/c$, respectively. The quantity λ is the incident wavelength and c is the light speed in vacuum. The UPML thickness is $10\Delta z$.

The accuracy of the CUDA implementation is verified by comparing the numerical results with those obtained by sequential execution on the CPU. Figure 7 demonstrates the bistatic scattering from an exponential soil surface with characteristic parameters $\delta = 0.1\lambda$ and

Table 4 Speedup improvement with asynchronous transfer.

f_i (GHz)	Mesh (Δ)	CPU time	GPU time	Speedup
1	4096	237.35	7.42	31.98x
4	16,384	3576.23	53.58	66.74x
8	32,768	14532.75	179.21	81.09x
16	65,536	59636.62	652.18	91.44x
32	131,072	241638.54	2595.31	93.10x
64	262,144	1039048.13	11144.73	93.23x

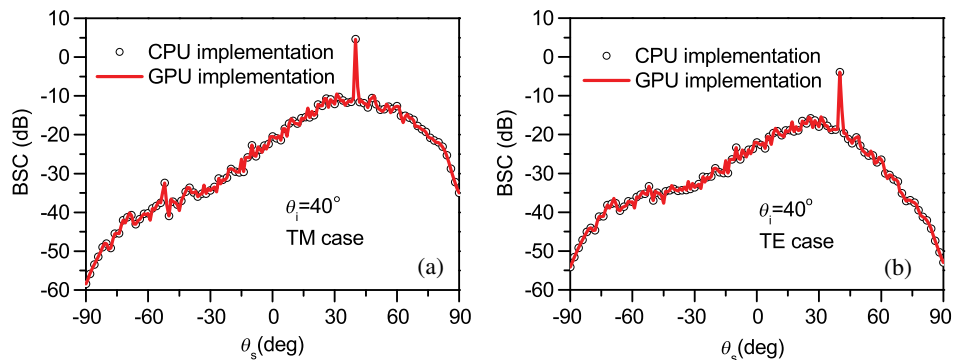


Fig. 7 Comparisons of the bistatic scattering from a soil surface by two implementations: (a) TM case and (b) TE case.

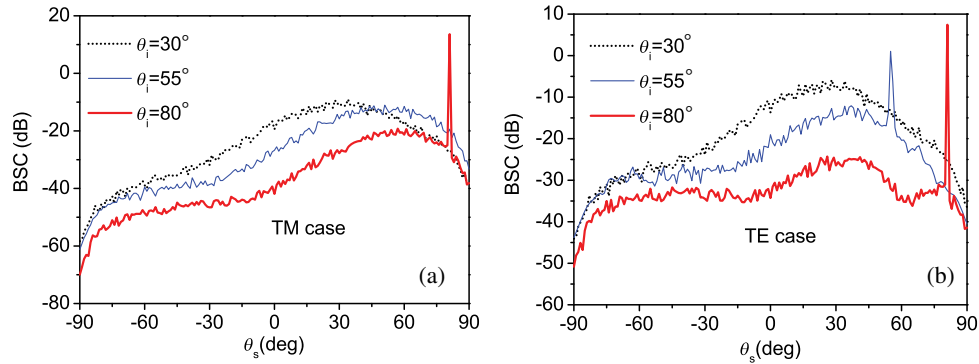


Fig. 8 The bistatic scattering from a rough surface under different incident angles: (a) TM case and (b) TE case.

$l = 1.0\lambda$ under the incident angle $\theta_i = 40$ deg at the incident frequency of $f_i = 1$ GHz. The generated length of the rough surface is $L = 204.8\lambda$ (4096Δ). The real and imaginary values of relative permittivity of the soil surface with 3.8% moisture are taken as $\epsilon_r = (2.5, 0.18)$.²² The results averaged by 20 surface realizations are in good agreement with the two implementations for both TM and TE incidences, demonstrating the accuracy of our FDTD-CUDA implementation. The times consumed for traditional FDTD schemes are approximately 88.25 and 91.23 min for the TM and TE cases, respectively. By contrast, the computation times of GPU-based FDTD are 2.59 and 2.43 min for the two incident cases. As is obvious, the time cost is dramatically reduced by the use of GPU implementation.

The scattering properties of a soil surface with length $L = 6553.6\lambda$ (131072Δ) for different incident angles increasing from the small incidence $\theta_i = 30$ deg to low grazing incidence $\theta_i = 80$ deg at the incident frequency of $f_i = 1.9$ GHz by the GPU-based FDTD implementation are investigated in Fig. 8. Here, the surface characteristic and electrical parameters are $\delta = 0.1\lambda$, $l = 1.0\lambda$ and $\epsilon_r = (2.5, 0.18)$ for both TM and TE cases, respectively. Scattering in the specular

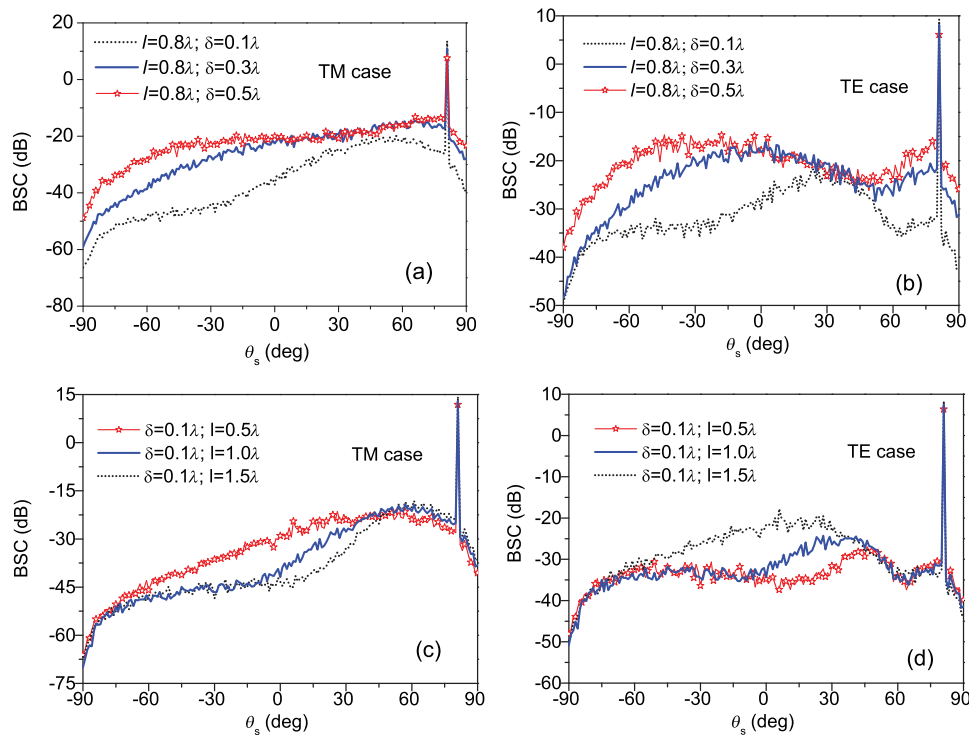


Fig. 9 The bistatic scattering from a rough surface with different characteristic parameters: (a) TM ($l = 0.8\lambda$; $\delta = 0.1\lambda$, $\delta = 0.3\lambda$, $\delta = 0.5\lambda$); (b) TE ($l = 0.8\lambda$; $\delta = 0.1\lambda$, $\delta = 0.3\lambda$, $\delta = 0.5\lambda$); (c) TM ($l = 0.5\lambda$, $l = 1.0\lambda$, $l = 1.5\lambda$; $\delta = 0.1\lambda$); and (d) TE ($l = 0.5\lambda$, $l = 1.0\lambda$, $l = 1.5\lambda$; $\delta = 0.1\lambda$).

direction is strongest for the grazing incident angle regardless of the polarization of the incident wave. It should be noticed that there is a specular peak in the case of the TM incidence, and when TE wave is studied, the scattering for the grazing incidence in the specular direction is also larger than that for small incident angles.

Figure 9 compares the influence of rough surface characteristic parameters including the correlation length l and the rms height δ on the EM scattering from 1-D large scale soil surface ($L = 6553.6\lambda$) under a low grazing incident angle $\theta_i = 80$ deg for our implementation. The incident frequency is $f_i = 5.9$ GHz.

Figures 9(a) and 9(b) plot the bistatic scattering coefficient versus the scattering angle θ_s with different rms heights $\delta = 0.1\lambda$, $\delta = 0.3\lambda$, and $\delta = 0.5\lambda$ keeping the correlation length $l = 0.8\lambda$ for TM and TE cases. For both TM and TE cases, the specular scattering decreases with the increase of rms height δ . Because the rms slope increases with increasing rms height, this leads to a decrease of the scattered energy in the coherent scattering direction. Figures 9(c) and 9(d) show the dependency of the bistatic scattering coefficient on the correlation l versus the scattering angle for TM and TE incidence waves. As shown in Figs. 9(c) and 9(d), with increasing correlation length, the specular scattering increases for both polarizations. The rms slope decreases with increasing correlation length resulting in stronger scattering in the specular direction for both TM and TE modes.

5 Conclusions

In this paper, the GPU implementation of the FDTD method is applied to investigate the EM scattering from a large scale rough soil surface with an exponential spectrum at the low grazing angle. Shared memory is utilized to optimize our implementation to improve the performance, and favorable speedup factors are achieved by comparing the computation time with that of sequential execution on CPU, which shows that the GPU-based FDTD has an obvious advantage in the study of large scale surfaces over the sequential CPU implementation. Finally, influences from incident angle, correlation length, as well as rms height on the bistatic scattering coefficient are also investigated and analyzed by the algorithm. When a target above or below a rough surface is studied, traditional high-frequency techniques are ineffective in handling the model. Therefore, future investigations on this topic will focus on the composite scattering from a two-dimensional target above a 1-D randomly rough surface using the GPU-based FDTD method.

Acknowledgments

This work was supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225002) and the Aeronautical Science Fund and Aviation Key Laboratory of Science and Technology on AISSS (Grant No. 20132081015).

References

1. M. Martorella, F. Berizzi, and E. D. Mese, "On the fractal dimension of sea surface backscattered signal at low grazing angle," *IEEE Trans. Antennas Propag.* **52**, 1193–1204 (2004).
2. H. C. Ku et al., "Fast and accurate algorithm for electromagnetic scattering from 1-D dielectric ocean surface," *IEEE Trans. Antennas Propag.* **54**, 2381–2391 (2006).
3. L. Tsang et al., "Electromagnetic computation in scattering of electromagnetic waves by random rough surface and dense media in microwave remote sensing of land surfaces," *Proc. IEEE* **101**, 255–279 (2013).
4. E. I. Torsos, "The validity of the Kirchhoff approximation for rough surface scattering using a Gaussian roughness spectrum," *J. Acoust. Soc. Am.* **83**, 78–92 (1988).
5. A. K. Sultan-Salem and G. L. Tyler, "Validity of the Kirchhoff approximation for electromagnetic wave scattering from fractal surfaces," *IEEE Trans. Geosci. Remote Sens.* **42**, 1860–1870 (2004).

6. L. X. Guo et al., "A high order integral SPM for the conducting rough surface scattering with the tapered wave incidence-TE case," *Prog. Electromagn. Res.* **114**, 333–352 (2011).
7. 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," *Prog. Electromagn. Res.* **89**, 149–166 (2009).
8. M. R. Pino et al., "The generalized forward-backward method for analyzing the scattering from targets on ocean-like rough surfaces," *IEEE Trans. Antennas Propag.* **47**, 961–969 (1999).
9. M. Y. Xia et al., "An efficient algorithm for electromagnetic scattering from rough surfaces using a single integral equation and multilevel sparse-matrix canonical-grid method," *IEEE Trans. Antennas Propag.* **51**, 1142–1149 (2003).
10. J. Li et al., "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* **26**, 1494–1502 (2009).
11. H. X. Ye and Y. Q. Jin, "Parameterization of the tapered incident wave for numerical simulation of electromagnetic scattering from rough surface," *IEEE Trans. Antennas Propag.* **53**, 1234–1237 (2005).
12. P. Sypek, A. Dziekonski, and M. Mrozowski, "How to render FDTD computations more effective using a graphics accelerator," *IEEE Trans. Magn.* **45**(3), 1324–1327 (2009).
13. W. W. Ma, D. Sun, and X. L. Wu, "UPML-FDTD parallel computing on GPU," in *Microwave and Millimeter Wave Technology (ICMMT), 2012 Int. Conf. on*, Vol. 4, pp. 1–4, IEEE, Shenzhen, China (2012).
14. M. Livesey et al., "Development of a CUDA implementation of the 3D FDTD method," *IEEE Antennas Propag. Mag.* **54**, 186–195 (2012).
15. A. K. Fung, M. R. Shah, and S. Tjuatja, "Numerical simulation of scattering from three-dimensional random rough surface," *IEEE Trans. Geosci. Remote Sens.* **32**, 986–994 (1994).
16. J. Li, L. X. Guo, and H. Zeng, "FDTD investigation on bistatic scattering from a target above two-layered rough surfaces using UPML absorbing conditioned," *Prog. Electromagn. Res.* **88**, 197–211 (2008).
17. S. D. Gedey, "An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices," *IEEE Trans. Antennas Propag.* **44**, 1630–1639 (1996).
18. S. D. Gedey, "An anisotropic PML absorbing media for the FDTD simulation for fields in lossy and dispersive media," *Electromagnetics* **16**, 399–415 (1996).
19. A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Boston (2005).
20. J. A. Kong, *Electromagnetic Wave Theory*, Wiley, New York (1986).
21. *NVIDIA CUDA C Programming Guide, Version 4.2*, NVIDIA Corporation, Santa Clara, California (2012).
22. J. Curtis, *Dielectric Properties of Soils: Various Sites in Bosnia (Data Rep.)*, US Army Corps of Engineers, Waterways Experiment, Washington, D.C. (1996).

Chungang Jia received a BS degree in 2009 from the School of Science, Taiyuan University of Technology, China, and he is currently pursuing a PhD degree at the School of Physics and Optoelectronic Engineering, Xidian University, China. His research interests include GPU high-performance computing in remote sensing and computational electromagnetics.

Lixin Guo received an MS degree in radio science from Xidian University, Xi'an, China, and a PhD degree in astrometry and celestial mechanics from Chinese Academy of Sciences, Beijing, China, in 1993 and 1999, respectively. During 2001 to 2002, he was a visiting scholar at School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu, Republic of Korea. His research interests mainly include: electromagnetic wave propagation and scattering in random media, and inverse scattering.

Ke Li received a BS degree in electronic information science and technology from Xidian University, Xi'an, China, in 2010. He is currently pursuing a PhD degree in radio science from the School of Physics and Optoelectronic Engineering, Xidian University. His research interests include the areas of computational electromagnetics.