DOI: 10.1051/epjconf/201611925018

MASSIVELY PARALLEL COMPUTATION OF SOIL SURFACE ROUGHNESS PARAMETERS ON A FERMI GPU

Xiaojie Li [1,3]*, Changhe Song[2]

[1]*Northeast Institute of Geography and Agroecology, Chinese Academy of Sciences, Changchun, 130102, China, *Email: lixiaojie@iga.ac.cn*

[2] *State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China*

[3]*Space Science and Engineering Center, University of Wisconsin, Madison, WI, 53706, USA*

ABSTRACT

Surface roughness is description of the surface micro topography of randomness or irregular. The standard deviation of surface height and the surface correlation length describe the statistical variation for the random component of a surface height relative to a reference surface. When the number of data points is large, calculation of surface roughness parameters is time-consuming. With the advent of Graphics Processing Unit (GPU) architectures, inherently parallel problem can be effectively solved using GPUs. In this paper we propose a GPU-based massively parallel computing method for 2D bare soil surface roughness estimation. This method was applied to the data collected by the surface roughness tester based on the laser triangulation principle during the field experiment in April 2012. The total number of data points was 52,040. It took 47 seconds on a Fermi GTX 590 GPU whereas its serial CPU version took 5422 seconds, leading to a significant 115x speedup.

Keywords: Soil surface roughness; Correlation length; Graphics Processing Unit (GPU); High Performance Computing (HPC)

1. INTRODUCTION

Surface roughness is a description of the randomness or irregularity of the microtopography. The standard deviation of surface height (σ) and the surface correlation length (*l*) describe the statistical variation for the random component of a surface height relative to a reference surface. Surface roughness forms a key factor to soil microwave radiation, while observing the earth using a microwave radiometer [1-2].

Several researchers focused on determination of surface roughness parameters. Benjamin [3] and Wang [4] studied satellite images for extraction of soil roughness parameters. A method using a roller chain was developed by Ali Saleh to measure soil roughness [5]. Oh [6] developed an inversion method of soil moisture and surface roughness from backscatter measurements of vegetation canopy. Tang [7] obtained soil roughness parameters through digital image processing method. McDonald developed a stereo vision system capable of measuring soil surfaces [8]. Also, the contact method was used by García [9] and Šařec [10]. Lidar technology is also using in the field of testing soil roughness [11-12]. Moreover, our lab presented a surface roughness testing that features rapid testing speed and high testing precision, requires no manual work and obtains 3D parameters for the surface [13]. It has the similar principle with Lidar, but easier and quicker. So they can share the data processing method. The total data points in the range of 40,000~100,000 can be obtained by a 500mm*600mm area. However, calculations of the soil roughness parameters based on these data points present a huge work and require a large computing time.

Graphics Processing Units (GPUs) possess hundreds of parallel processor cores and are capable of executing tens of thousands of threads in parallel. GPUs have been used very successfully for many different computational problems [14-15]. In this paper we propose a GPU-based massively parallel computing method for 2D bare soil surface roughness estimation. We reached a speedup factor of 115x as compared to the single threaded CPU version when GPU I/O transfer is taken into account.

2. METHODOLOGY

2.1 Formulas of soil surface roughness

The standard deviation of a surface height (σ) and the surface correlation length (l) are defined as follows [16]:

a). Standard Deviation of a Surface Height:
Consider a surface in the *x-y* plane whose height at a point *(x, y)* is *z(x, y)* above the *x-y* plane. For a statistically representative segment of the surface, of dimensions Lx and Ly, the standard deviation σ for the discrete one-dimensional case is given by

$$\sigma = (\overline{z^2} - \bar{z}^2)^{1/2} = [\frac{1}{N-1}(\sum_{i=1}^{N}(z_i)^2 - N(\bar{z})^2]^{1/2} \quad (1)$$

where $\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i$ and N is the number of samples.

b). Surface Correlation Length
The normalized autocorrelation function for a one-dimensional surface profile z(x) is defined as

$$\rho(x') = \frac{\int_{-L_x/2}^{L_x/2} z(x)z(x+x')dx}{\int_{-L_y/2}^{L_y/2} z^2(x)dx} \quad (2)$$

In our program, because the result produced by the microwave radiometer and radar detector, the scattering values are spread over a 2D area. So, we want to calculate the 2D correlation coefficient. First we need to calculate the distance between each pair of data points as follows,

$$r = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \quad i,k = 1,2,\dots N \quad (3)$$

where $x' = (j-1)\Delta x$, j is an integer larger than 1.

Take a Δx, with each value of j, find x' . Calculate Equation (2) to get all the points (the total is M) where $r = x'$, x' being the spatial distance between two points. Considering the normalized data point, equation becomes

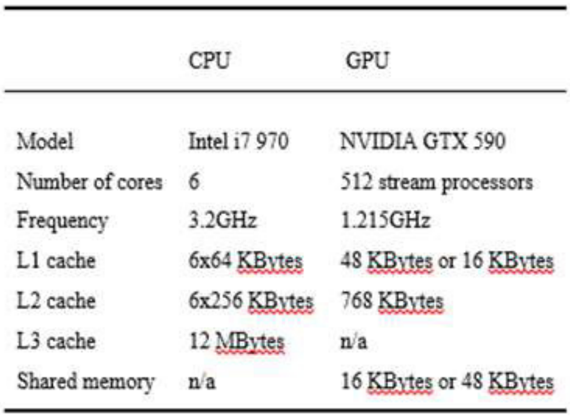
$$\rho(j) = \frac{\sum_{i=1,k=1}^{M}(z_i - \bar{z}_i)(z_k - \bar{z}_k)|_{r=(j-1)\Delta x}}{\sum_{i=1}^{M}(z_i - \bar{z}_i)^2 |_{r=(j-1)\Delta x}} \quad (4)$$

The surface correlation length l usually is defined as the displacement x' for which $\rho(x')$ is equal to 1/e, i.e. $\rho(l) = 1/e$. The unit of l is mm.

2.2 GPU Hardware Specification and CUDA Computing Basics

We use CUDA to implement a GPU-based surface roughness parameter computation scheme because of its easy deployment and high computing performance [17]. Moreover, it is best suited to NVIDIA GPU implementations [17-18] as compared to the other computing techniques such as OpenCL and Direct Compute. The hardware specifications of one NVIDIA GTX 590 GPU employed in our study are depicted below

Table 1. Device parameters of the GPU and CPU



	CPU	GPU
Model	Intel i7 970	NVIDIA GTX 590
Number of cores	6	512 stream processors
Frequency	3.2GHz	1.215GHz
L1 cache	6x64 KBytes	48 KBytes or 16 KBytes
L2 cache	6x256 KBytes	768 KBytes
L3 cache	12 MBytes	n/a
Shared memory	n/a	16 KBytes or 48 KBytes

2.3 CUDA Implementation

As 3D tester, the data includes 3 parts, x-coordinate, y-coordinate and z-coordinate. First of all, we need write C codes. Once correct C codes were obtained, in the conversion from C codes to CUDA codes, some caution need to be taken.
In our implementation, 300 different distances are thus chosen. As the CUDA architecture automatically schedules the execution in different blocks, the number of threads in one block, i.e. the block size, is however optimized experimentally. For the influence of the thread block size on the computing performance, we carry out a test to vary the block size. The results are presented in Figure 1.

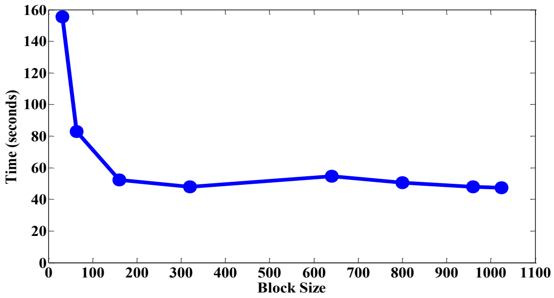


Figure 1. Time usage for 52,040 data points with different block sizes

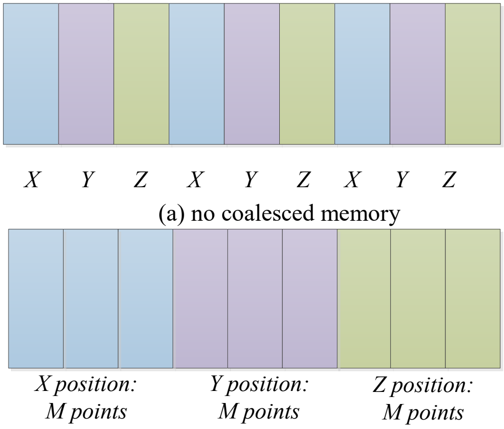
From Figure 1, we can observe that the best performance is achieved when the block size is 1024. Hence, we set the block size to be 1024. The shared memory usage is consequently determined.

A command "cudaFuncCachePreferL1" was launched in our CUDA codes, which means that the usage of L1 cache is larger than shared memory.

2.4 The other Considerations

1). Memory Coalescing

The input data is a 3D data array stored in a text file. The first column is x-coordinate, the second column is y-coordinate, and third is z-coordinate. For coalesced memory access, the array is stored column wise in the memory, which means the x-coordinates of all points are stored together in the memory, followed by all the y-coordinates and then all the z-coordinates of all points, that is shown in Figure 2.



(a) no coalesced memory

(b) coalesced memory

Figure2. A pictorial layout of the data structure.

2). Coalesced memory access

The coalesced memory access technique combines several memory accesses together so as to save the memory bandwidth. In our study, contiguous data will be accessed by neighboring threads due to our designed data layout. This access pattern makes coalesced memory access possible. However, in the case of two-dimensional data, the length of the first dimension is required to be a multiple of warp size in order to achieve full efficiency during coalesced memory access. Therefore, we pad the number of points to be a multiple of the warp size.

3). Shared Memory

We used shared memory to save the local sum of two local variables. When all threads are finished, the local sums are buffered to the shared memory. Then the local sums in the shared memory are accessed and summed up again by each of these threads.

3. RESULTS

In order to validate this method, we used the field test date to calculate, which was carried out at the Jilin Agricultural University, China in April 19, 2012. The pictures of bare soil and the original data by the surface roughness tester are below.

Figure 3. (a) Bare soil picture.

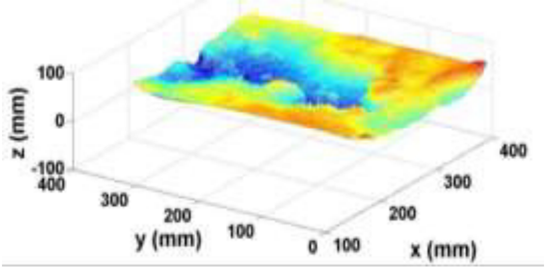


Figure 3. (b) The original data obtained by 3D tester

The computation is originally implemented in C programming language using gcc as the compiler. We used gcc -lstdc++ correlation.c -O3 for C codes. The CUDA codes were compiled using nvcc (NVIDIA CUDA compiler) version of 5.5 and executed on NVIDIA GTX590 GPU with compute capability of 2.0. The compiler options thus are –arch=compute_20, code=sm_20 -fmad=false -ftz=true. The timing results calculated at different data points using CPU and GPU are shown in Table 2 along with the speedup for the GPU based implementation. The correlation length of 49.09 mm is obtained in this case. The standard deviation of the surface height is 12.70 mm.

Table 2. Timing results for the CPU and GPU

Data points	Time usage (unit:s)		Speedup
	CPU	GPU	
52040	5422	47	115

4. CONCLUSIONS

In this paper, we developed a GPU-based parallel computing method for 2D surface roughness estimation. It was written in CUDA and was executed on an NVIDIA GeForce GTX 590 GPU. Calculating the distance between any two points and estimating whether this distance is an assignment value can be carried out by threads. In our implementation, we designed the computations for different appointed distances such that they are performed in different blocks. This method was applied to the data collected by the surface roughness tester on April 2012 based on laser triangulation principle. For a total of 52,040 data points, the serial CPU version of the implementation took 5422 seconds whereas the GPU implementation took 47 seconds, leading to a significant 115x speedup.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China Nos. 41201335 at Space Science and Engineering Center, University of Wisconsin-Madison USA. The authors thank Professor Bormin Huang for his help.

REFERENCES

[1] Sungwook, H, 2010: Surface roughness and polarization ratio in microwave remote sensing, *International Journal of Remote Sensing*, **31(10)**, 2709-2716.

[2] Zheng, X. M., Zhao, K, 2010: A method for surface roughness parameter estimation in passive microwave remote sensing. *Chinese Geographical Science*, **20(4)**, 345-352.

[3] Benjamin, S., Leonie, D. F., Jo-Ann, H.; Felix, W., 2010: Fast derivation of soil surface roughness parameters using multi-band SAR imagery and the Integral Equation Model. *2010 International Conference on Pattern Recognition*, IEEE computer society, 3931-3934.

[4] Wang, S. G., Li, X, Han, X. J., Jin, R., 2011: Estimation of surface soil moisture and roughness from multi-angular ASAR imagery in the Watershed Allied Telemetry Experimental Research (WATER). *Hydrology and Earth System Sciences*, **15**, 1415-1426.

[5] Ali Saleh., 1993: Soil roughness measurement: Chain method. *Journal of Soil and Water Conservation*. **48(6)**, 527-529.

[6] Oh, Y., Hong, J. Y., Jung, S. G., 2007: Retrieval of Soil Moisture and Surface Roughness from Backscatter Measurements of Vegetation Canopy. *Asia-Pacific Microwave Conference*, 1-3.

[7] Tang, X. J., Xiao, H., Ding, H., Liu, J. H., 2009: Surface roughness measurement based on image processing and image recognition. *Computers and Simulation in Modern Science*. 91-96.

[8] McDonald, A. J., Crossley, S., Bennett, J. C, Cookmartin, G., Morrison, K., Quegan, S., 1999: Stereo vision measurements of soil surfaces and their utility in testing microwave backscattering models. *Geoscience and Remote Sensing Symposium*, **5**, 2634 - 2636.

[9] Garcia Morenoa, R., Diaz Alvarezb, M.C., Saa Requejob, A., Valencia Delfac, A.M., 2010: Tarquisb Multiscaling analysis of soil roughness variability. *Complexity and Nonlinearity in Soils*.**160(1)**, 22–30.

[10] Šařec, P., Šařec, O., Prošek, V., Čížková, K., 2007: Laser profilometer testing by laboratory measurements. *RES. AGR. ENG.*, **53 (1)**, 1-7.

[11] Russell T., Rocco, P., Mihai A. Tanase, Kim L., Jorg M. Hacker, Jeffrey P. Walker. 2014: Estimation of soil surface roughness of agricultural soils using airborne LiDAR. *Remote Sensing of Environment*, **140**, 107-117.

[12] Shi, X., Girod, L., Long, R., DeKett, R., Philippe, J., Burke, Tom. 2012: A comparison of LiDAR-based DEMs and USGS-sourced DEMs in terrain analysis for knowledge-based digital soil mapping. *Geoderma*, **170,** 217-226.

[13] Li, X. J., Zhao, K., Zheng, X. M., 2012: Development of surface roughness tester based on laser triangulation method. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE)*, **28(8)**, 116-121.

[14] Sui, H.G., Peng, F. F., Xu, C., Sun, K.M., Gong, J.Y., 2012: GPU-accelerated MRF segmentation algorithm for SAR images. *Computers & Geosciences*, **43**,159-166.

[15] Devrim, A., Pakize, E.□ 2015: GPU accelerated training of image convolution filter weights using genetic algorithms. *Applied Soft Computing*, **30**, 585-594.

[16] Ulaby, F. T., Moore, R. K., Fung, A. K., 1987: Microwave remote sensing active and passive. *Radar remote sensing and surface scattering and emission theory*, **2**, 819-833.

[17] Mielikainen, J., Huang, B. M., Huang, H. L. A, Goldberg, M. D., 2012: Improved GPU/CUDA Based Parallel Weather and Research Forecast (WRF) Single Moment 5-Class (WSM5) Cloud Microphysics. *Applied Earth Observations and Remote Sensing*, **5(4)**, 1256-1265.