A Morphological LiDAR Points Cloud Filtering Method based on GPGPU

Shuo Li[1], Hui Wang[1], Qiuhe Ma[1] and Xuan Zha[2]
[1]Zhengzhou Institute of Surveying & Mapping, No.66, Longhai Middle Road, Zhengzhou, China
[2]Tianjin Institute of Hydrographic Surveying and Charting, No.40, Youyi Road, Tianjin, China

Keywords: LiDAR, Morphology, Points Cloud Filtering, GPU, CUDA.

Abstract: Because of its large amount of data, airborne LiDAR points cloud filtering is often time-consuming. On the basis of the traditional morphological LiDAR points cloud filtering, a method which adopted the parallel technique based on GPU and assigned the massive operations to be parallel executed in many computing unit to achieve the purpose of fast filtering was proposed. Through the corresponding experiments, the validity and efficiency of the proposed LiDAR points cloud filtering method were verified.

1 INTRODUCTION

GPGPU (General Purpose Computing on Graphics general Processing Units) is proposed by SIGGRAPH (Special Interest Group for Computer GRAPHICS) in 2003. GPGPU applies heterogeneous computing resources for large-scale parallel computing, rather than merely make use of GPU for general computing (Qiu, 2011). NVIDA launched a new general computing product of GPU in 2007, CUDA (Compute Unified Device Architecture), deeply influencing the development of GPGPU and graphics.

The traditional serial algorithm is improved based on CUDA C language, effectively reducing the running time of the algorithm with the full use of the powerful parallel computing capability and floating point data processing speed of GPU.

LiDAR (Light Detection and Ranging) has a broad development prospect in the field of remote sensing, because it can obtain the 3D information of the space surface rapidly. LiDAR is a remote sensing technology that has developed rapidly in recent years. It can realize the synchronous, rapid and precise acquisition of the spatial 3D coordinates, providing a simple and effective technique for the rapid achievement of spatial information (Zhu, 2006).

LiDAR data filtering is a process of removing non-ground points from laser points data and extracting digital elevation model (DEM) (Zhang, 2007). Existing filtering algorithms can be divided into four categories: slope-based, the smallest block, surface-based and clustering/seg-mentation (Sithole, 2004). The morphology method belongs to the surface-based methods. It was proposed by Lindenberger of University of Stuttgart in 1993 and was improved by many scholars subsequently. Kilian (1996) added the weight factors in the method, and Zhang (2003) proposed a progressive morphological filtering method based on gradually increasing the window size of the filtering and using elevation difference thresholds. It is notable that filtering and the corresponding quality control are the most pivotal and time-consuming step in the subsequent progress of LiDAR to generate DEM, accounting for approximately 60% to 80% of the processing time (Sithole, 2004). Therefore, rapid implementation of LiDAR points cloud filtering becomes an urgent problem to be solved, when the amount of data is large.

To solve this problem, Sui (2010), Li (2011) and Zhang (2009) improved the existing morphology filtering methods to filter rapidly. However, the improved algorithms and the original algorithms are serial, of which calculation unit is CPU. A morphology filtering method is proposed in this paper based on GPGPU, effectively improving the efficiency of the subsequent processing of LiDAR data.

Li, S., Wang, H., Ma, Q. and Zha, X.
A Morphological LiDAR Points Cloud Filtering Method based on GPGPU.
In *Proceedings of the 2nd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2016)*, pages 80-84
ISBN: 978-989-758-188-5

2 MORPHOLOGY FILTERING

The proposed method mainly includes three key steps: establishing the virtual grid based on the original discrete points cloud, morphology filtering based on virtual grid and judging the attributes of all points. The specific principles are as follows.

2.1 Establishing the Virtual Grid

The establishment of virtual grid is not the interpolation of the raw data, but building a virtual grid capable of indexing to each discrete point. The point is assigned to the corresponding grid according to the plane coordinate. The calculation formula is as follows:

$$\begin{cases} I = (X - X_{\min}) / c \\ J = (Y - Y_{\min}) / c \\ c = 1 / \sqrt{n} \end{cases} \tag{1}$$

c: sampling interval; n: number of points; $X_{\min}$: the minimum abscissa; $Y_{\min}$: the minimum ordinate; (X, Y) : coordinates of the discrete point.

2.2 Morphology Filtering

Mathematical morphology method for extracting image information has mature theory basis. It is based on two basic operations, dilation and erosion. Open operation and close operation are two different combinations of the two operations.

The digital surface model of airborne laser data can also be processed based on mathematical morphology. If the coordinate of a grid point p is (x, y, z) , the dilation of z is as follows:

$$d_p = \max_{(x_p, y_p) \in W} (z_p) \tag{2}$$

W: window size; (x_p, y_p, z_p) : coordinates of a point in the neighbourhood of point p.

The window of the method is a 2D square window. The output of the dilation is the maximum elevation.

Similarly, the formula of the erosion is as follows:

$$d_p = \min_{(x_p, y_p) \in W} (z_p) \tag{3}$$

The result of the erosion is the minimum elevation. Open operation is the dilation after erosion, and close operation is the erosion dilation.

Mathematical morphological filtering of laser data is the opening operation in essence. Firstly, the discrete points of trees and objects which are smaller than the window size are removed based on erosion. Secondly, the boundaries of the objects which are bigger than window size are restored according to dilation.

2.3 Judging the Attributes of All Points

The elevation difference between the point in the virtual grid and the representative point is compared with the threshold to determine whether the point has the same attribute with the representative point. All points are processed in this way to distinguish the ground points from the non-ground points.

3 ACCELERATION STRATEGY BASED ON GPGPU

As is shown in Figure 1, the proposed algorithm is a parallel improvement of the traditional serial LiDAR points cloud filtering method. The figure is divided into CPU part and GPU part by the dotted line. The data processing steps above the dotted line are executed in CPU and steps below the dotted line are implemented in GPU. The transmission block indicates the data transmission between the host memory and the GPU video memory and the corresponding arrow manifests the direction of data transmission.

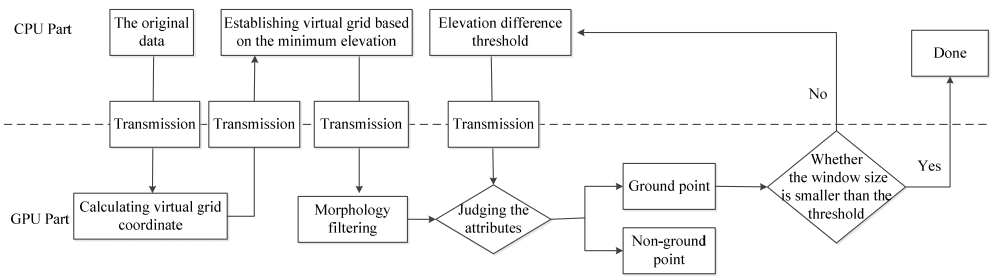


Figure 1: The flow of morphology filtering based on GPGPU.

3.1 Acceleration Strategies of Establishing Virtual Grid and Judging Attributes of All Points

The process to establish virtual grid has the following characteristics:

- The processing flows of all points are the same. The virtual grid coordinate of each point is calculated as formula (1).
- The points are irrelevant. The calculation of one point is independent of the results of other points.
- The amount of data is large. The amount of points cloud is generally millions or even more.

The above characteristics are in line with the SIMT (Single Instruction Multiple Threads) model of CUDA. The process can be improved to be parallel, the specific strategy are as follows:

After the original points cloud is read in the memory, the discrete points cloud data is copied to the GPU video memory so that the calculation unit of GPU can access the original data. The data is assigned to different threads to be processed respectively by the program based on SIMT model, making use of multiple calculation units of GPU to achieve the purpose of parallel calculation. In other words, the gird coordinate of the point is calculated in parallel by all calculation units of GPU.

The process of judging the attributes of all points is similar to the above calculation, so it can be accelerated in the same way.

3.2 Acceleration Strategy of Morphology Filtering

The elapsed time of morphology calculation accounts for more than 50 percent of all the filtering time in traditional serial algorithm. The acceleration of morphology calculation is a key step to the implementation of rapid filtering. The improvement strategies of morphology filtering are as follows:

- The virtual grid data is bound to texture memory to accelerate the accessing in the GPU part. The size of the whole virtual gird data is compared to the maximum size of 2D texture to which GPU is bound. If the size of the virtual grid data is bigger than the maximum size, the virtual grid data will be processed in the divided blocks. A block data will be stored to the video memory and bound to 2D texture in GPU part at a time. And all results will be merged into a whole after all the block data is calculated. If the size of the virtual grid data is smaller than the maximum size, all virtual grid data will be stored to the video memory and bound to 2D texture.
- Erosion of data bound to texture memory is the first step in the process of morphological opening operation. The output is stored to the global memory and bound to texture memory. Then dilation is implemented to achieve the result of the open operation.
- The erosion and dilation of points are assigned to different threads to be processed for the purpose of parallel calculation. Thread organization mode should be set according to the actual hardware environment to ensure the high utilization of computing resources.
- The function "_syncthreads" provided by CUDA is used to ensure thread synchronization, judging whether all open operations in the area are executed.
- The window size is increased in GPU part according to the window size and elevation difference threshold acquired from CPU part. The process is iterated to judge the attributes of the points until the parameters meet the stop condition.

4 EXPERIMENTS

4.1 Experimental Environment

The experimental hardware environments are Intel i5 dual core 2.9GHz CPU, 8G memory and 1G video memory of ASUS GTX 570 graphics card. The experimental software environments are Windows7 operating system, Microsoft Visual Studio 2010 development environment, C++ programming language and CUDA 4 parallel development kit.

The experimental data is 15 sample data provided by ISPRS. The statistic of error rate of filtering algorithm is based on the referential filtering result of the corresponding sample.

4.2 Results and Analysis

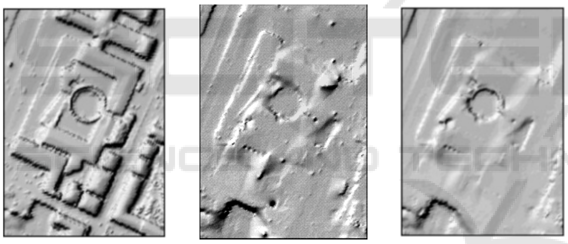
4.2.1 Evaluation of Filtering Results

The experimental data is points cloud data of Vaihingen/Enz and Stuttgart captured by ALTM of Optech, provided by the Commission Ⅲ of ISPRS in 2003. The former nine samples (Sample11-Sample42) are urban data, of which the point distance is 1-1.5m. The latter six samples (Sample51-Sample71) are mountain data, of which the point distance is 2-3.5m. The data is subdivided into several types (Chen, 2007).

The results of three samples are shown in Figure 2, Figure 3 and Figure 4 respectively. Figure 2(a), Figure 3(a) and Figure 4(a) are three terrain maps based on the original data. Figure 2(b), Figure 3(b) and Figure 4(b) are three terrain maps based on filtering results of the proposed algorithm. Figure 2(c), Figure 3(c) and Figure 4(c) are three terrain maps based on the referential filtering results. It is obvious that the proposed algorithm is capable of filtering the objects in the samples, such as houses, plants, bridges and so on.

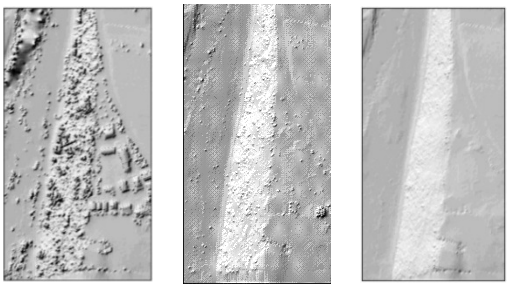
ISPRS group has adopted 8 filtering methods based on 15 samples (Lee, 2003). The results and the result of the proposed algorithm are shown in table 1. The first row in the table 1 is the names of methods. The error rates of the results are below the names (Huang, 2009).

Compared with the 8 classical algorithms, the total error rate of the proposed algorithm is stable on the whole and ranks high. This fully shows that the proposed algorithm can adapt to various terrain. The reliable filtering proves the effectiveness of the proposed algorithm.



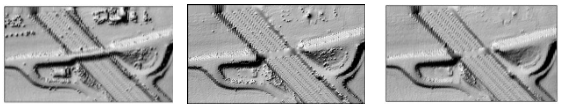
(a): Sample23. (b): Sample23. (c): Sample23.

Figure 2: The filtering results of houses.



(a): Sample51. (b): Sample51. (c): Sample51.

Figure 3: The filtering results of plants.



(a): Sample71. (b): Sample71. (c): Sample71.

Figure 4: The filtering results of bridges.

4.2.2 Comparison of Filtering Efficiency

The elapsed time reflects the efficiency of filtering. When processing the same data, the shorter the elapsed time, the higher the efficiency.

In order to verify the efficiency of the proposed algorithm, the proposed algorithm based on GPGPU is compared with the traditional serial morphological algorithm. The data is several LiDAR points cloud data of different size. As shown in table 2, the ten samples are sorted from small to large according to the number of points. The statistical results are shown in table 2 and the corresponding elapsed time is shown in Figure 5.

As is shown in Table 2 and Figure 5, the elapsed time of the traditional serial algorithm increases linearly with the amount of the LiDAR points cloud growing up and faster than that of the proposed

Table 1: Total Error Rates of Nine Algorithms.

Sample	Total Error Rate (%)									Rank
	Elmvist	Sohn	Axelsson	Pfeifer	Brovelli	Roggero	Wack	Sithole	The Proposed Algorithm	
Samp11	22.40	20.49	10.76	17.35	36.96	20.80	24.02	23.25	13.80	2
Samp12	8.18	8.39	3.25	4.50	16.28	6.61	6.61	10.21	6.77	5
Samp21	8.53	8.80	4.25	2.57	9.30	9.84	4.55	7.76	1.96	1
Samp22	8.93	7.54	3.63	6.71	22.28	23.78	7.51	20.86	6.81	3
Samp23	12.28	9.84	4.00	8.22	27.80	23.20	10.97	22.71	7.67	2
Samp24	13.83	13.33	4.42	8.64	36.06	23.25	11.53	25.28	10.23	3
Samp31	5.34	6.39	4.78	1.80	12.92	2.14	2.21	3.15	1.86	2
Samp41	8.76	11.27	13.91	10.75	17.03	12.21	9.01	23.67	9.42	3
Samp42	3.68	1.78	1.62	2.64	6.38	4.30	3.54	3.85	2.11	3
Samp51	23.31	9.31	2.72	3.71	22.81	3.01	11.45	7.02	6.89	4
Samp52	57.95	12.04	3.07	19.64	45.56	9.78	23.83	27.53	20.36	5
Samp53	48.45	20.19	8.91	12.60	52.81	17.29	27.24	37.07	14.14	3
Samp54	21.26	5.68	3.23	5.47	23.89	4.96	7.63	6.33	3.34	2
Samp61	35.87	2.99	2.08	6.91	21.68	18.99	13.47	21.63	4.26	3
Samp71	34.22	2.20	1.63	8.85	34.98	5.11	16.97	21.83	3.73	3

Table 2: Acceleration Rate.

Sample	Number of points (Thousand）	The elapsed time of the proposed parallel algorithm(s)	The elapsed time of the serial morphological algorithm(s)	Acceleration rate
1	100	0.078	0.109	1.4
2	200	0.094	0.203	2.2
3	300	0.109	0.297	2.7
4	500	0.156	0.500	3.2
5	1000	0.210	1.060	5.0
6	2000	0.390	2.100	5.4
7	3000	0.480	2.900	6.0
8	4140	0.630	4.493	7.1

algorithm. The acceleration rate stays stable after a sharp climb. Two primary reasons are as follows:

- The traditional serial algorithm is implemented in CPU. Since CPU has only one calculation unit, so the elapsed time will be longer if the amount of data increases. On contrary, GPU has several calculation units so that the elapsed time of the proposed algorithm increases more slowly than that of the traditional serial algorithm.
- The computing resources are not used fully when the amount of data is small. Thus, the acceleration rate is low. The larger the amount of data, the more the resources are used and the higher the acceleration rate. The acceleration rate stays stable when the resources are fully used.

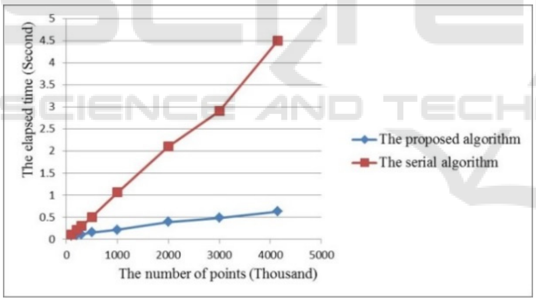


Figure 5: Comparison of the elapsed time.

5 CONCLUSIONS

GPGPU is applied in morphology filtering method for the purpose of parallel filtering. The proposed method of morphological LiDAR points cloud filtering can remove non-ground points effectively and better meet the filtering requirements. Moreover, it is a reliable and efficient LiDAR point cloud filtering method, which has certain practical value.

REFERENCES

Qiu, D., Y., 2011. *GPGPU Programming*. China Machine Press，Beijing, China, pp. 9-10.

Zhu, S., C., 2006. The Technique Principle of LiDAR and Its Application in Surveying and Mapping. *Modern Surveying and Mapping*, 4(7), pp. 12-13.

Zhang, X., H., 2007. *The Theory and Method of Measuring Technology of Airborne Laser Radar*. Wuhan University press, Wuhan, China, pp. 42-43.

Sithole, G., Vosselman, G., 2004. Experimental Comparison of Filter Algorithms for Bare Earth Extraction from Airborne Laser Scanning Point Clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(1), pp. 85-101.

Kilian, J., Haala, N., Englich, M., 1996. Capture and Evaluation of Airborne Laser Scanner Data. *International Archives of Photogrammetry and Remote Sensing*, 31(B3), pp. 383-388.

Zhang, K., Q., Chen, S., C., Whitman, D., et al 2003. A Progressive Morphological Filter for Removing Nonground Measurements from Airborne LiDAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4), pp. 872-882.

Sui, L., C., Zhang, Y., B., Liu, Y., et al, 2010. Filtering of Airborne LiDAR Point Cloud Data Based on the Adaptive Mathematical Morphology. *Acta Geodaetica et Cartographica Sinica*, 4(39), pp. 390-395.

Li, P., C., Wang, H., Liu, Z., Q., et al, 2011. A Morphological LiDAR Points Cloud Filtering Method Based on Scan Lines. *Journal of Geomatics Science and Technology*, 28(4), pp. 274-277.

Zhang, Y., B., Sui, L., C., Qu, J., et al, 2009. Fast Filtering of Airborn LiDAR Point Cloud Data Based on Mathematical Morphology. *Bulletin of Surveying and Mapping*, 5, pp. 16-19.

Chen, Q., Gong, P., Baldocchi, D., et al, 2007. Filtering Airborne Laser Scanning Data with Morphological Methods. *Photogrammetric Engineering and Remote Sensing*, 73(2), pp. 175-185.

Lee, H., S., Younan, N., H., 2003. DTM Extraction of LiDAR Returns via Adaptive Processing. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9), pp. 2063-2069.

Huang, X., F., Li, H., Wang, X., et al, 2009. Filter Algorithms of Airborne LiDAR Data: Review and Prospects. *Acta Geodaetica et Cartographica Sinica*, 38(5), pp. 466-469.