

A Fast Planner Detection Method in LiDAR Point Clouds Using GPU-based RANSAC

Jun Lan¹, Yifei Tian², Wei Song³, Simon Fong⁴, and Zhitong Su⁵

¹ North China University of Technology - P.O. Box 100144, Beijing, China
junlan1234@163.com

² University of Macau - P.O. Box999078, Macau China
tianyifei0000@sina.com

³ North China University of Technology - P.O. Box 100144, Beijing, China
Corresponding: sw@ncut.edu.cn

⁴ University of Macau - P.O. Box999078, Macau China
ccfong@umac.mo

⁵ North China University of Technology - P.O. Box 100144, Beijing, China
suzhitong@ncut.edu.cn

Abstract Unmanned ground vehicle (UGV) technology benefits for intelligent transportation and automatic digital map reconstruction in smart city domain. In most urban areas, the planes are usually existed in buildings and infrastructures, which consist of massive 3D points detected by the sensors on UGV. This paper aims to study planar detection from 3D point clouds and represent the planes using a mesh instead of plenty of points in 3D reconstruction. A fast planar detection method using GPU-based Random Sample Consensus (RANSAC) is developed to estimate the planar parameters existed in surrounding environment. The sensed 3D point clouds are segmented into several planes. Each planar surface is reconstructed and rendered using only four vertices, so that the memory usage is reduced in 3D reconstruction. For improving the time efficiency in plane recognition process, GPU is utilized to fasten RANSAC algorithm using CUDA parallel programming technology.

Keywords: *Unmanned ground vehicle, RANSAC, Terrain modeling, GPU, LiDAR*

1 Introduction

Citizen-centered urban construction provides a visible service and data visualization interfaces for smart city, which focuses on the artificial intelligent applications for modern social lifestyles [13]. For realizing urban environment representation, street view map is a traditional method by rendering the images captured at the fixed positions, which lacks of 3D information [14]. Sensor-carried unmanned ground vehicle (UGV) technology is researched to collect environment information and percept surrounding terrain [3]. In order to realize autonomous driving, UGV is required to analyze the percept environment information and make corresponding decisions to avoid obstacle automatically [6]. Path planning of UGV in unknown environment are decided based on obstacle perception result of the surrounding information collected from several different sensors, such as stereovision [10,7]. The environment information collected from such sensors is prone to be affected by the illumination situation, which is not enough to support the decision making of UGV in intelligent transportation [9]. Kinect is other depth sensor to capture environment information but the detection precision does not satisfy the required accuracy of UGV [17]. Light Detecting And Ranging (LiDAR) is utilized to obtain high-precision 3D point cloud for restoring surrounding terrain models quickly and accurately [16]. With plenty of planes at different positions and orientations existed in most urban areas, plane representation is considered as an ideal part in 3D reconstruction, which is rendered by a mesh plane instead of plenty of sensed 3D points. Besides, with the plane detection result, environment perception of UGV becomes easily and convenient to be realized. This way, a Random Sample And Consensus (RANSAC) algorithm is proposed to recognize plane model from large-scale point cloud [11]. However, LiDAR point cloud is dispersed and non-structured so that traditional RANSAC algorithm

requires massive iterations in plane detection. In order to improve the accuracy and speed performance of environment perception of UGV, this paper proposed a fast planner detection method in LiDAR point clouds using GPU-based RANSAC algorithm. In our method, raw 3D point cloud is segmented into several sub-spaces to detect planes. To recognize plane in individual sub-spaces, a preset plane model is initialized to record the point count of which suitable to the plane model. After test several different plane model, the plane model of the most registered points is considered as the plane in the sub-space. Thus, the optimum plane model in each sub-space is computed in parallel by a CUDA development kit. The remainder of this paper is organized as follows, Section 2 overviews related work. Section 3 discusses the method of plane detection. Section 4 describes the experiments using the proposed method. Section 5 concludes this paper.

2 Related Works

In mobile robot and intelligent vehicle domain, plane recognition methods based on 3D point clouds are widely researched in 3D environment sensing technology. Ishida et al. [5] detected planes from depth pixels by using a 3D Hough transform algorithm in unknown environment. In the method, the Hough space was divided into several parts to extract plane parameters through several votes and iterations. Hulik et al. [4] utilized a 3D Hough transform to extract large planes from point cloud collected from LiDAR sensor. In order to preserve the accuracy of the plane detection process, a Gauss smoothing function was used to eliminate the noise distribution in the Hough space after executing Hough transform on LiDAR points. To increase the speed performance of updating process in Hough space, a caching technique was applied in point registration. Compared with 2D Hough transform algorithm, 3D Hough transform utilized more time consumption in plane detection process. The detection precision of plane parameters is not accurate enough to support the decision making of UGV so that 3D Hough transform always utilized in indoor environment. In 3D environment scene, the RANSAC algorithm is widely researched to extract parameters from LiDAR point cloud. Schnabel et al. [12] utilized tradition RANSAC algorithm to compute plane parameters from three random and dispersed points. Through several iterations, a goal function was utilized to search the optimum plane model with the most fitting counts of the remaining points. To improve operational efficiency of RANSAC, Choi et al. [1] reduced unknown number of plane equation model. Wang et al. [15] proposed a preprocessing model based on bucketing mode. Nevertheless, RANSAC algorithm is hard to achieve fast speed due to its iterative computation characteristics. CPU is well suited for complex and intensive computing tasks. However, it has been difficult to improve the computational efficiency of the RANSAC algorithm implemented on large-scale datasets. In the past few years, GPU have transformed from a graphics-rendering device to a high-performance computing device that solves the parallelism problems [8,2]. Data-intensive computing tasks perform more efficiently on large-scale datasets using GPU. Utilize the inherent parallelism of RANSAC and the high computational efficiency of the GPU, this paper proposes a fast planner detection method in LiDAR point clouds.

3 PLANAR DETECTION METHOD

To estimate the plane parameters from the raw 3D point cloud accurately, a plane detection framework is proposed as shown in Figure 1. The framework mainly is consist of three procedures, including the segmentation of 3D point cloud, mapping the point cloud to the GPU, plane detection and plane fitting.

3.1 Point Cloud Segmentation

It is difficult to extract plane fromLiDAR points cloud because the distribution characters of uneven density and discontinuous spatial. Firstly, raw 3D point clouds sensed from 3D space are divided to some subspaces. The 3D point count existed in each subspace is assumed as D_{numi} . D_{nummax} is the maximum count in all 3D point clouds. The changing rate of 3D point cloud density between adjacent spaces is formulated using the equation (1):

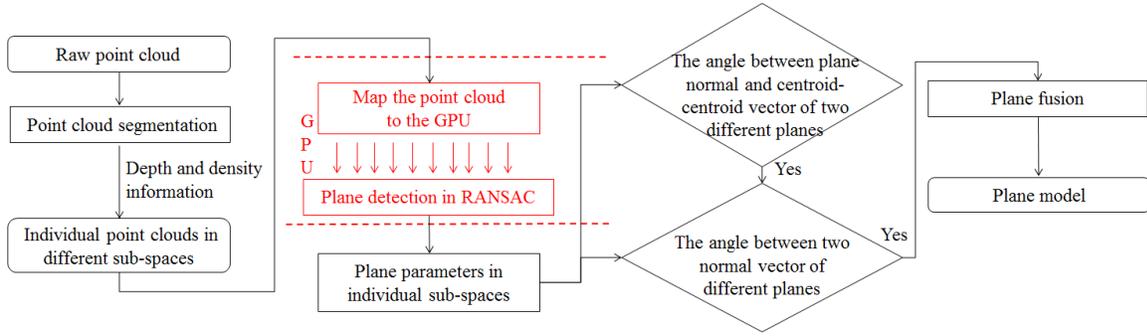


Figure 1: Framework of the proposed planner detection method in LiDAR point clouds using GPU-based RANSAC

$$D_{den_changei} = \frac{D_{deni+1} - D_{deni}}{D_{nummax}} \quad (1)$$

The entire space of 3D point cloud is divided into a few subspaces H_i by maximum and minimum of z-axis. After establishing the statistical histogram of change rate in point cloud density, 3D point clouds are merged if changing rate of 3D point cloud density $D_{den_changei}$ is lower than a threshold α .

$$D_{den_changei} \leq \alpha, H_i \cup H_{i+1} \quad (2)$$

3.2 Plane Detection

After the 3D point clouds are segmented into several subspaces and registered into the terrain model on the CPU, each data in spatial is processed independently. Because GPU has many blocks and there are many threads in each block. In order to improve computing efficiency of algorithm, the registered 3D point cloud datasets are copied to the GPU memory, which are able to be executed in parallel. The 3D points in each subspace P_i are executed four procedures as follow:

1. Plane L' is constructed by three point selected from P_i , the normal vector of the plane L' is n' .
2. Angle α_i is calculated as the angle between n' and the other points in P_i . Score SL' of plane L' records the number of angles less than a threshold.
3. After repeating 1),2) for T times, plane L' with the highest score is chosen. T is formulated using the equation (3):

$$T = \frac{\log(1 - \mu)}{\log(1 - (1 - \tau)^2)} \quad (3)$$

The variable μ is the chances of best plane to be selected. The variable τ is the proportion outside the point of plane L' .

4. Marking plane L' and removing these points.

3.3 Plane Fitting

If the difference between the normal vectors of two planes is less than threshold θ , expressed as equation (4), we assume they are coplanar:

$$\arccos(n_i, n_j) \leq \theta \quad (4)$$

In the other plane fitting constraint, centroid-centroid vector M_{ij} is the line vector calculated from the plane L_i centroid to the plane L_j centroid. When the angles between centroid-centroid vector and the planes' normal vectors n_i or n_j are approximately vertical to each other, the two planes L_i and L_j are considered as belong to a same plane. Thus, if the angles between the two vectors less than a threshold φ after subtracting 90-degrees, the plane fitting constraint is satisfied as shown in equation (5):

$$\begin{aligned} \arccos(M_i, n_i) - 90 &\leq \varphi \\ \arccos(M_j, n_j) - 90 &\leq \varphi \end{aligned} \quad (5)$$

The variable φ is nearly to zero.

4 EXPERIMENTS

In this section, we analyze the performance of the proposed plane detection method from LiDAR points. The experiments were implemented on a 3.20 GHz Intel® Core™ Quad CPU computer with a GeForce GT 770 graphics card, 4 GB RAM. The applied HDL-32E Velodyne LiDAR had capability to sense 32×12 3D points in a packet per 560.96 μ s. In our experiment, the 3D point clouds were built using DirectX software development kits. Figure 2 presents the raw datasets of $180 \times 32 \times 12$ points tested by a stationary LiDAR. In the experiment, all the sensed 3D points were divided into 20 groups according to their y coordinates. The changing rate of 3D point cloud in density histogram is shown in Figure 3. Then, using a threshold to merge point cloud based on the histogram, some redundant points were removed as shown in Figure 4. Finally, a series of initial plane models was calculated after initialization detection. In Figure 5, the optimum plane was detected after plane fitting.

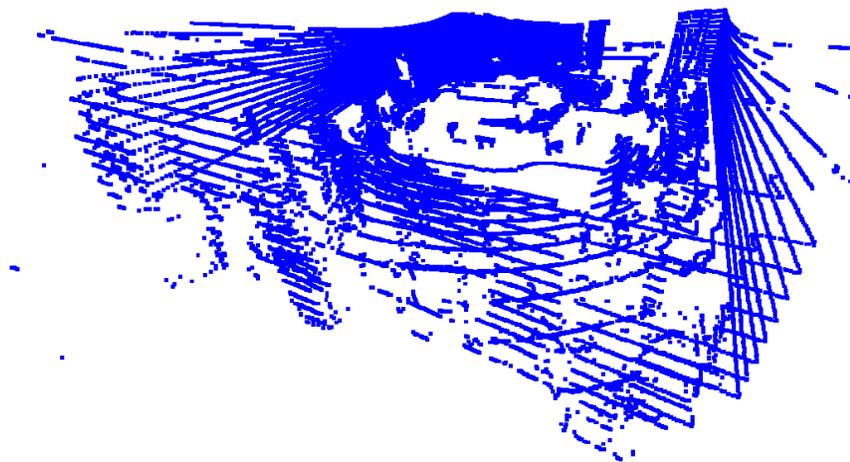


Figure 2: The datasets of LiDAR point clouds

To demonstrate the performance, we compared the proposed method with traditional RANSAC [5] and CPU-optimized RANSAC plane detection method [4]. The experiments were carried on a dataset of 2520 points, whose processing speeds were shown as in Table 1. The traditional RANSAC algorithm could

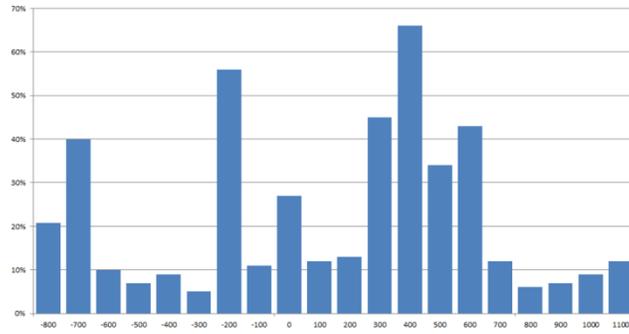


Figure 3: The changing rates of the density histogram



Figure 4: A segmentation result of the 3D point cloud

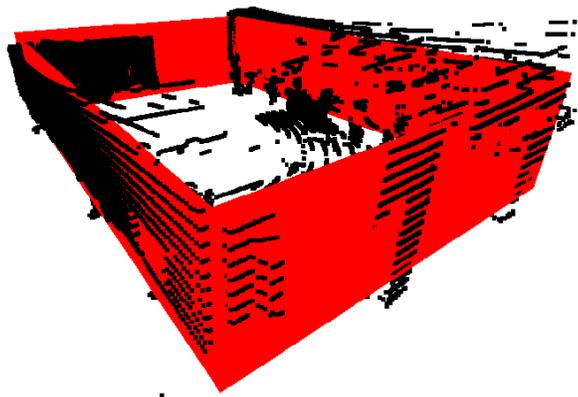


Figure 5: A planar detection result

not perform in real time. Final detection results of CPU-optimized and GPU plane detection methods were shown as in Figure 6 and Figure 7. It was obvious that our proposed method was implemented fast and accurately.

Table 1: Result of Efficiency test experiments

Method	Number of point clouds	Comp. Time
RANSAC Algorithm	2520	>3 min
CPU-optimized RANSAC plane detection method	2520	1 min 30s
Plane detection using GPU	2520	2s

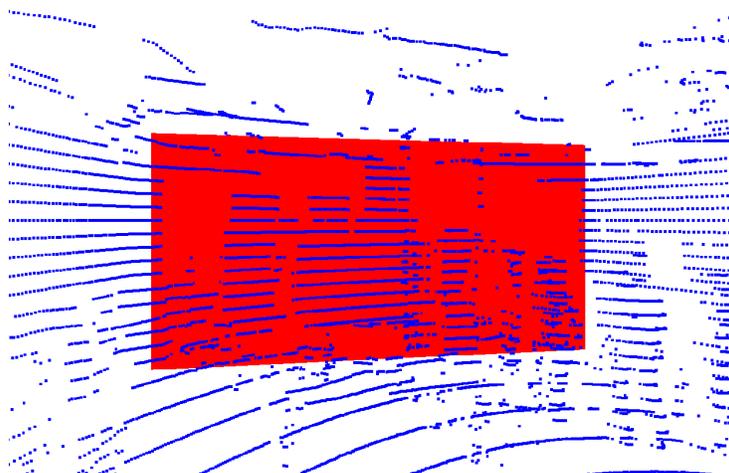


Figure 6: CPU-optimized plane detection method

5 CONCLUSIONS

To contribute efficient planar detection method for urban terrain modeling, this paper demonstrated a GPU-based RANSAC method in LiDAR point clouds. The point segmentation and plane fitting processes reduced iteration times. Using GPU programming technology, the improved RANSAC method performed planar detection much faster than CPU-based method. CPU executed the complex computation processes, while GPU executed the interactive and repeated processes of low computation complexity. This way, the proposed method combined the advantages of CPU and GPU. The proposed method was able to reduce the memory consumption for the urban terrain reconstruction for smart city modeling and intelligent transportation. In the future, we will apply this plane detection method in the 3D terrain modeling on UGV.

ACKNOWLEDGMENTS This research was supported by the National Natural Science Foundation of China (61503005), by NCUT “The Belt and Road” Talent Training Base Project, by NCUT “Yuxiu” Project, by Beijing Natural Science Foundation (4162022), and by High Innovation Program of Beijing (2015000026833ZK04).

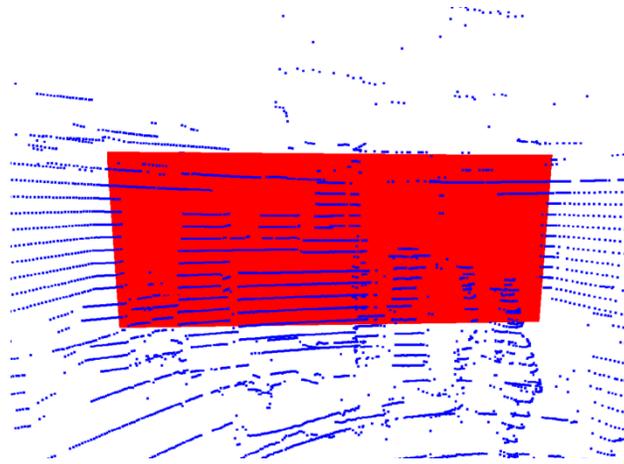


Figure 7: Plane detection using GPU

References

1. S. Choi, J. Park, J. Byun, and W. Yu. Robust ground plane detection from 3d point clouds. In *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pages 1076–1081, Oct 2014.
2. N. Faujdar and S. Saraswat. A roadmap of parallel sorting algorithms using gpu computing. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 736–741, May 2017.
3. X. Feng, E. S. Dawam, and S. Amin. A new digital forensics model of smart city automated vehicles. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 274–279, June 2017.
4. Rostislav Hulik, Michal Spanel, Pavel Smrz, and Zdenek Materna. Continuous plane detection in point-cloud data based on 3d hough transform. *J. Vis. Comun. Image Represent.*, 25(1):86–97, January 2014.
5. Y. Ishida, H. Izuoka, H. Chinthaka, N. Premachandra, and K. Kato. A study on plane extraction from distance images using 3d hough transform. In *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, pages 812–816, Nov 2012.
6. K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo. Development of autonomous car-part i: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, 61(12):7131–7140, Dec 2014.
7. Z. Khalid, E. A. Mohamed, and M. Abdenbi. Stereo vision-based road obstacles detection. In *2013 8th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–6, May 2013.
8. Alexander Kubias, Frank Deinzer, Matthias Kreiser, and Dietrich Paulus. Efficient computation of histograms on the gpu. In *Proceedings of the 23rd Spring Conference on Computer Graphics, SCCG '07*, pages 207–212, New York, NY, USA, 2007. ACM.
9. Ki Yong Lee, Joon Woong Lee, and Myeong Rai Cho. Detection of road obstacles using dynamic programming for remapped stereo images to a top-view. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 765–770, June 2005.
10. Caio César Teodoro Mendes and Denis Fernando Wolf. Real time autonomous navigation and obstacle avoidance using a semi-global stereo method. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 235–236, New York, NY, USA, 2013. ACM.
11. P. Rizwan, K. Suresh, and M. R. Babu. Real-time smart traffic management system for smart cities by using internet of things and big data. In *2016 International Conference on Emerging Technological Trends (ICETT)*, pages 1–7, Oct 2016.
12. R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.
13. Erfan Babae Tirkolae, Ali Asghar Rahmani Hosseinabadi, Mehdi Soltani, Arun Kumar Sangaiah, and Jin Wang. A hybrid genetic algorithm for multi-trip green capacitated arc routing problem in the scope of urban services. *Sustainability*, 10(5):1–21, 2018.

14. A. Torii, M. Havlena, and T. Pajdla. From google street view to 3d city models. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2188–2195, Sept 2009.
15. Xiaoyan Wang, Hui Zhang, and Sheng Liu. Reliable ransac using a novel preprocessing model. *Computational and mathematical methods in medicine*, 2013, 2013.
16. S. Yang and Y. Fan. 3d building scene reconstruction based on 3d lidar point cloud. In *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pages 127–128, June 2017.
17. Hyun Woo Yoo, Woo Hyun Kim, Jeong Woo Park, Won Hyong Lee, and Myung Jin Chung. Real-time plane detection based on depth map from kinect. In *IEEE ISR 2013*, pages 1–4, Oct 2013.