지형분할을 위한 다채널 라이다 데이터 처리

푸옹[1], 조성재[1], 심성대[2], 곽기호[2], 조경은[1]*
[1]동국대학교 멀티미디어공학과
[2]국방과학연구소
*e-mail : cke@dongguk.edu (교신저자)

Multi-channel Lidar Processing for Terrain Segmentation

Phuong Chu[1], Seoungjae Cho[1], Sungdae Sim[2], Kiho Kwak[2], Kyungeun Cho[1]*
[1]Department of Multimedia Engineering, Dongguk University-Seoul
[2]Agency for Defense Development

요 약

In this study we propose a novel approach to segment a terrain in two parts: ground and none-ground. The terrain is gained by a multi-channel 3D laser range sensor. We process each vertical line in each frame data. The vertical line is bounded by the sensor's position and a point in the largest circle of the frame. We consider each pair of two consecutive points in each line to find begin-ground and end-ground points. All points placed between a begin-ground point and an end-ground point are ground ones. The other points are none-ground. After examining all vertical lines in the frame, we obtain the terrain segmentation result.

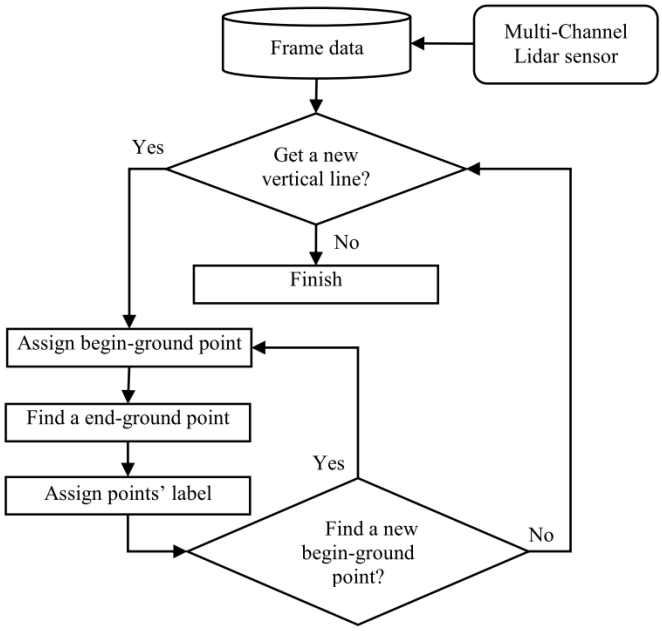
1. Introduction

In our study, we focus on terrain segmentation which is an important thing for any autonomous moving system. Our target is to divide a point cloud into two groups: ground points and non-ground points. The autonomous vehicle has possibility to move in the none-ground area such as buildings, walls. Especially, if the terrain has an enormous gradient, the vehicle has no ability to go through. Therefore, we also put all points in this area to non-ground group.

There are some previous studies about ground segmentation but they have some disadvantages. The method [1] only works well in flat urban environment and it is not effective for slopped terrain. In [2] the authors give another method which is very simple and easy to implement for solving problem in non-flat urban environment. This method is relied on a graph and normal vector estimation. This method is not effective when we apply for the sparse data with many holes. In [3] and [4], the authors propose other ideas by dividing a point cloud data set in smaller part like voxel or block. After processing, they combine all small part to gain the result. But the processing time is long. The paper [5] represents a method which is closest with ours. However, the implementations are different. In our method, we divide each frame of point cloud into small groups called vertical line. In each line, we find all special points and assign label "ground" or "none-ground" for each point.

2. Terrain segmentation algorithm

The terrain segmentation system is demonstrated in Fig. 1. We obtain and process each frame data from a Multi-Channel Lidar sensor in local coordinate. The original coordinate is always sensor's position. As we mentioned above, we process each vertical line. If we use m-channel Lidar sensor, we usually have m+1 points in each line.



(Fig. 1) Terrain segmentation framework.

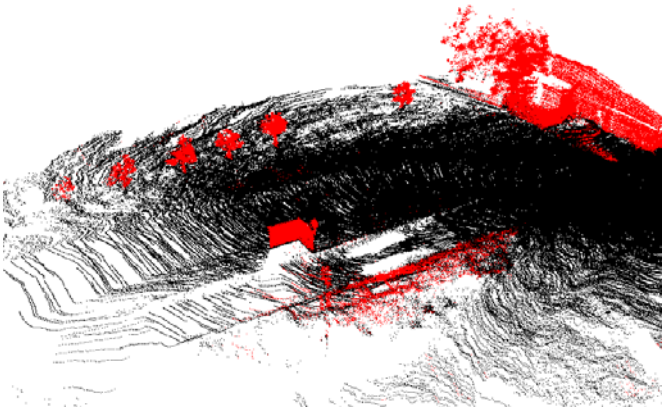
In each vertical line, we will find all begin-ground points and end-ground points. After that, we assign label for all points in the line. All points placed between a begin-ground point and an end-ground point are ground ones. The other points are none-ground. The first begin-ground point is always ground point at sensor's position even if this point is not in point cloud data. From this point we find the first end-ground point.

After a none-ground point, if we detect a point which is

lower than previous point, it could be a new begin-ground called potential point. To guarantee, we calculate the height difference between potential point and the nearest end-ground point. If the height difference is smaller than *h_min*, the potential point is a new begin-ground point. Otherwise, we assign "none-ground" label for previous point and try to find a new begin-ground point by consider current and next points.

3. Experiments and analysis

The terrain For our implementation, the multi-channel Lidar sensor is a Velodyne HDL-32E. The sensor returns approximately 60.000 points in each frame. We employed a PC with Intel Core i5-4690 CPU (3.5 GHZ) and 8GB RAM. All experiments show good results in both flat and none-flat terrains. Fig. 2 shows an example of the result. The black and red points are ground and none-ground, respectively. The average processing time of our algorithm is 4.7 milliseconds.



(Fig. 2) Result of terrain segmentation.

4. Conclusion

In this paper, we proposed a novel approach for terrain segmentation using a multi-channel Lidar sensor. The core idea of our algorithm is to process each vertical line in local coordinate. In addition, we consider the slope, lost points and abnormality of the distance between two consecutive points to detect ground and none-ground points. The experiments show that our method achieves good results in many terrains. Moreover, for approximately 60.000 points, the average processing time is only 4.7 milliseconds. In future work, we will enhance the algorithm for segmentation of bumpy terrain.

Acknowledgements

This research was supported by BK21 Plus project of the National Research Foundation of Korea Grant and was supported by a grant from Agency for Defense Development, under contract #UD150017ID.

Reference

[1] J. Hernández, and B. Marcotegui, "Point Cloud Segmentation towards Urban Ground Modeling," 2009 Urban Remote Sensing Event, pp. 1-5, 2009.

[2] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D Lidar Data in non-flat Urban Environments using a Local Convexity Criterion," IEEE Intelligent Vehicles Symposium, pp. 215-220, 2009.

[3] S. Cho, J. Kim, W. Ikram, K. Cho, Y. Jeong, K. Um, and S. Sim, "Sloped Terrain Segmentation for Autonomous Drive Using Sparse 3D Point Cloud," The Scientific World Journal, vol. 2014, 2014.

[4] X. Lin,and J. Zhanga, "Segmentation-based ground points detection from mobile laser scanning point cloud," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015 International Workshop on Image and Data Fusion, pp. 99-102, 2015.

[5] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the Segmentation of 3D LIDAR Point Clouds," IEEE International Conference on Robotics and Automation, pp. 2798-2805, 2011.