Ground Segmentation Algorithm Based on 3D Lidar Point Cloud

Ziyang Cheng[a], Guoquan Ren, Yin Zhang

Department of Vehicle and Electrical Engineering, Army Engineering University, Shi Jiazhuang 050003, China.

[a]Czy_aeu@163.com

Keywords: 3D lidar, ground segmentation, line segment feature, real-time.

Abstract. Aiming at the problem of accurately and efficiently segmenting the ground from the 3D Lidar point cloud, a ground segmentation algorithm based on the features of the scanning line segment is proposed. The algorithm performs denoising and pose correction on the 3D point cloud and divides the scan line according to the Euclidean distance and vertical height difference between adjacent points. Then analyze the characteristics of the adjacent line segments, such as pitch, slope, and height difference, and mark them as the ground segments and obstacle segments according to the classification rules. Finally, the comparison experiments show that this algorithm can accurately segment the ground in real time.

1. Introduction

In the outdoor environment, the ground segmentation of the 3D Lidar data is an important pre-processing task for the local environment perception of the UGV (unmanned ground platform), and it is the basis for obstacle detection, classification, and dynamic target tracking. The main task of ground segmentation is to distinguish ground points and positive obstacles above the ground from the 3D point cloud acquired by Lidar, so it is possible to further identify obstacle types, and identify the Point set of obstacles after segmentation as pedestrians vehicles and vegetation, etc. In the process of driving, the platform will produce changes in pitch, roll, and suspension. At the same time, due to the uneven distribution of 3D Lidar data, the laser measurement points near the Lidar are relatively densely distributed, and the distribution of laser measurement points farther away from the Lidar is comparatively sparse, the space between measurement points is large, and in the case of processing millions of laser points, the accuracy and speed of the segmentation are difficult to balance. Based on the above reasons, making use of 3D Lidar data to detect accessible ground and various types of environmental obstacles and generating navigation point cloud maps has become a challenging task.

2. Related Work

As an active sensor, Lidar is not easily affected by the environment. Many research groups use it as the main sensor for UGV. Lidars at home and abroad have developed many point cloud segmentation methods based on Lidar. In the United States DARPA Unmanned Vehicle Challenge, the commonly used method of ground point cloud segmentation is to project the 3D point cloud onto a horizontal grid, and determine the lidar point attributes by comparing the difference in the height of the point cloud within the adjacent grid with a threshold value [1].

Kamme [2] proposed a ground detection algorithm based on the maximum height difference in the grid point cloud. If the maximum height difference in the grid is less than a certain threshold, it is marked as a ground area. Since the grid height difference method only calculates the height difference between the laser spots, the efficiency is high, but the under-segmentation problem easily occurs and the ground plane and positive obstacles cannot be well separated.

Moosman [3] first established the four-domain undirected graph of the point cloud, then fits the local plane based on the neighborhood information of the point and uses the local convexity of the plane normal vector to segment the point cloud, which has better segmentation results for different objects. However, the calculation of the convexity feature of the normal vector of the rough surface is not good.

Documents [4] [5] use the region growing algorithm for point cloud segmentation, which is easy to implement and fast. However, different regional growth strategies often result in decomposition results at different levels of detail.

The clustering-based segmentation algorithm regards the segmentation of the point cloud model as a data point classification process with certain geometric feature parameters. The literature [6]-[9] uses the surface meta-categories, Mean Shift clustering, and spectral clustering, respectively. K-means clustering and fuzzy clustering achieve the segmentation of the point cloud model, and the point cloud is divided into groups by feature similarity detection. These methods can produce stable results, but different clustering criteria will produce different clustering results, which limits the applicable objects of a cluster segmentation method, and improper clustering algorithm design may lead to over-segmentation or under-segmentation.

Hernandez and Marcotegui [10] proposed a method for segmenting well in a flat urban environment based on distance image hole filling algorithm, but this method is not suitable for slope terrain. Himmelsbach [11] constructed polar coordinate grid maps, established a non-parametric ground model based on straight line fitting in the fan-shaped region, and distinguished ground points and obstacle points by comparing the lidar points with the model.

In general, a good point cloud segmentation algorithm must solve the following key issues: (1) whether the selected features can be detected under a wide range of observation conditions, with higher sensitivity and less affected by noise; (2) How Correctly group all spatial neighbors belonging to a class; (3) How to effectively handle large-scale data. At present, methods including model fitting, clustering, regional growth, and global optimization have more or less room for improvement in these areas. In order to solve these problems, this paper proposes a point cloud segmentation method based on line segment features. The selected features are simple and easy to extract, and can accurately and efficiently segment the point cloud.

3. Point Cloud Segmentation

In the lidar point cloud, it can be seen that there is a clear difference between the line type and the line spacing by observing the distribution of ground points and obstacle points. The lidar points in the ground area are ring-shaped and the distance between the lines gradually increases away from the origin of the lidar. The lidar points in the obstacle area are linearly distributed. Therefore, this paper realizes the segmentation of point cloud by analyzing the distance between line segments, vertical height difference and slope.

3.1 Segment Segmentation

In order to make the points on each scan line the same kind of lidar points, we first perform line segmentation. Enter the laser points on a single scan line in sequence to calculate the Euclidean distance between adjacent points. Since the difference in vertical height ΔH between ground points and obstacle points is significantly different, it is divided into two cases: (1) In the case where the vertical height difference between adjacent points is smaller than the threshold Th, if the distance R between the current point and the previous point is less than the threshold Tr, it is determined to belong to the same scan line segment; if the distance is greater than the threshold Tr, the previous point is Marked as the ending point of the segment, the current point is marked as the starting point of the new segment. (2) When the vertical height difference between adjacent points is greater than the threshold Th, the previous point is marked as the ending point of the line segment, and the current point is marked as the starting point of the new line segment. With this strategy, even low road arbitrage can be accurately distinguished from the ground.

The scanning frequency of the Lidar is 10 Hz, and the angular resolution is at most 0.18°. Therefore, the distance between adjacent scanning points is theoretically:

$$R = D \times 0.18\pi/180 \tag{1}$$

Take 1.5 times as the value of the threshold Tr in the calculation.

3.2 Point Cloud Division

For flat terrain, the ground scan line spacing is usually larger than the obstacle scan line spacing, and the vertical height of obstacle points is often significantly larger than the ground point. We calculate the radial distance difference of the laser spot at the same scanning angle on the adjacent line segments and take the average value as the distance *d* between adjacent line segments. When it is greater than the given threshold T_d, it is determined that the line segment is a ground scan line, and vice versa is an obstacle scan line.

However, the slope topography will cause the point cloud ring to be compressed so that the distance between adjacent rings is less than the distance between the rings on a flat terrain, so there is a possibility of being misjudged as an obstacle. Therefore, we introduce the characteristics of the slope between segments, assuming that the segment contains the vertical height of the laser spot *H*, and the gradient *G* is defined as:

$$G = \left(\sum_{k=1}^{k} H_k \Big/ k - \sum_{n=1}^{n} H_n \Big/ n\right) \Bigg/ \left(\sum_{k=1}^{k} D_k \Big/ k - \sum_{k=1}^{k} D_k \Big/ k\right) \tag{2}$$

That is, the line segment contains the ratio of the average height difference of the laser spot to the average distance difference, and *k* and *n* are the points contained in the line segment. When the inclination between the line segment and the upper adjacent line segment is greater than the small threshold $T_{g\text{-}s}$ and less than the large threshold $T_{g\text{-}l}$, the sloped terrain is determined.

In addition to the terrain slope, the unmanned platform will roll and pitch during driving, resulting in the compression or unfolding of the Lidar scan line. For example, tilting the platform to the left will cause the floor scan line spacing on the left side of the vehicle to become smaller, thereby mistakenly detecting the flat ground as an obstacle. When the platform is lifted, the scan line spacing increases, and the obstacle may be mistakenly detected as ground. Therefore, the detection based on the scanning line spacing is very one-sided. This paper defines the distance threshold between the line segment and the upper adjacent line segment as a function of the distance value *D* returned by the lidar point and the vehicle roll angle β, instead of a fixed value. The empirical formula for the Spacing threshold function is:

$$T_d = \mu \times D \times \tan\beta \tag{3}$$

μ is a scale factor, which is a value of 0.018. The vehicle roll angle β is acquired by inertial navigation information. Therefore, if the vehicle is rolling to the left, the spacing threshold of the left scan line segment decreases as the ring approach vehicle. In this way, ground and obstacles can be reliably detected even in the case of roll and pitch of the sensor.

3.3 Algorithm Flow

According to the characteristics between the scan line segments, this method compares the different line segments in the left and right direction of the same scan line and the corresponding line segments in the front and rear scan lines. First, calculate the average distance from each point on the scan line segment to the lidar origin as the distance from the line segment to the lidar origin, then search from the minimum height scan line segment closest to the lidar origin: in the left and right direction, take the minimum height line segment as the initial ground line segment. If the height difference between other segment and it exceeds the threshold Th, the other segment is marked as an obstacle line segment, otherwise it is marked as ground line segment. In the fore-and-aft direction, the difference d between the current line segment and the previous line segment is compared. If it is less than T_d , the current line segment is marked as the ground line segment, otherwise it is the obstacle line segment. In the scan line segment marked as the ground, if the gradient G between the current line segment and the previous line segment is within the range of the large and small threshold, it is re-marked as the slope line segment. The algorithm flow is shown in Figure 1:

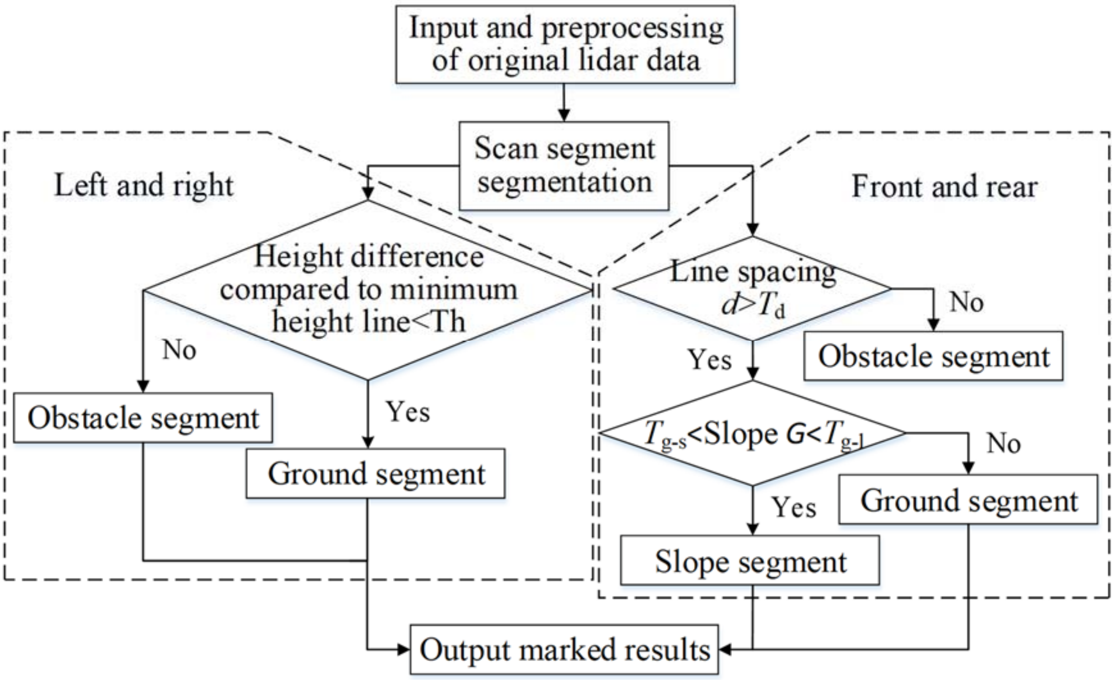
Input and preprocessing of original lidar data
Scan segment segmentation
Left and right
Front and rear
Height difference compared to minimum height line<Th
No
Yes
Obstacle segment
Ground segment
Line spacing $d>T_d$
No
Obstacle segment
Yes
$T_{g\text{-}s}$<Slope G<$T_{g\text{-}l}$
No
Ground segment
Yes
Slope segment
Output marked results

Fig 1. Algorithm flowchart

4. Experimental Process and Results

In order to verify the above method, This article collects data through a Velodyne HDL 32-line lidar with scanning frequency of 10 Hz installed on the ground unmanned experimental platform shown in Figure 2. An outdoor scene as shown in Figure 3 was selected for experiment, flat road ahead and 18° slope on the left. In experiment, T_h=0.3m, $T_{g\text{-}s}=\tan 10^{\circ}$ (0.18), $Tg_{\text{-}l}=\tan 30^{\circ}$ (0.58).

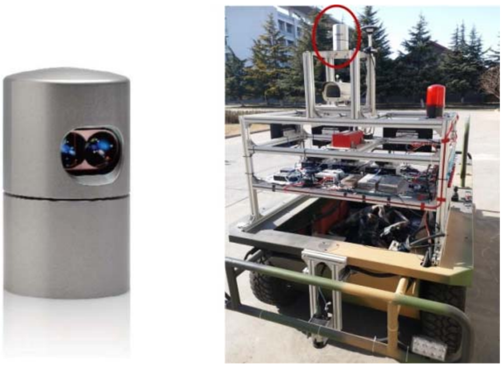

Fig 2. Ground unmanned experimental platform

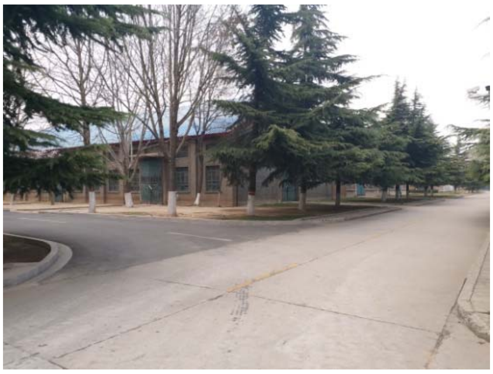

Fig 3. Test scenario

Figure 4 shows the results of the lidar point cloud processing. The scan line segments in the figure are marked green (ground segment) and red (obstacle segment), (a) is the segmentation result of the method in this paper, and (b) of local convex segmentation algorithm. It can be seen from the comparison that this method can better divide the flat road and slope , and effectively distinguish the obstacles above the road surface also even the low marginal roads; the local convexity algorithm uses the four adjacent points of the lidar point to fit the plane, the road there is no obvious change in the plane normal vector at the marginal roads, so ground misdetection occurs.

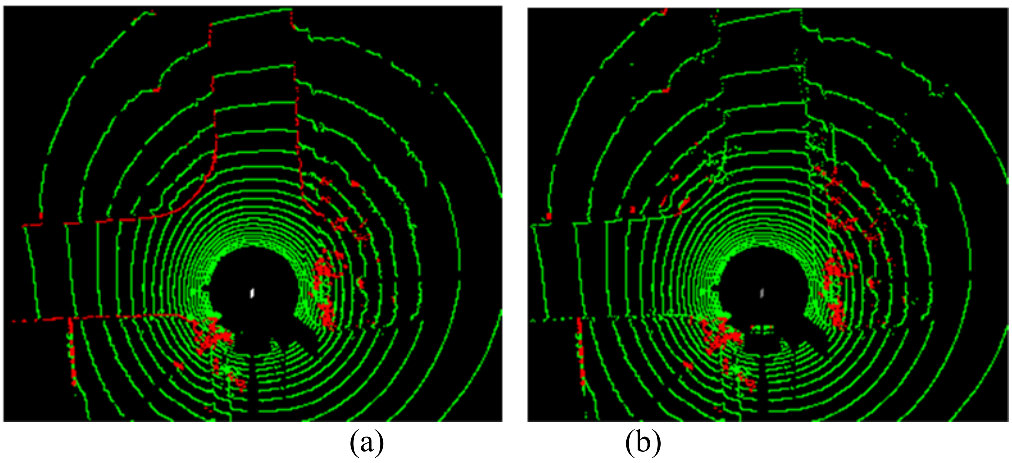
(a) (b)

Fig 4. point cloud segmentation results

The algorithm is based on an industrial computer equipped with an Intel Core i7 4GHz CPU and is implemented in C++. The running time is 24ms per frame, which is significantly lower than the 69ms of the local convexity algorithm and meets the requirement of real-time performance.

5. Conclusion

This paper presents a 3D Lidar ground segmentation algorithm based on scanning line segment features. The algorithm first segmented the ridar scan line segment, then the line segment was marked as the ground and the obstacle line segment by analyzing the characteristics such as the distance, inclination and vertical height between line segments. Finally, the experiment was conducted on a flat road surface. The results show that the algorithm has higher real-time performance and accuracy than the local convexity algorithm and can segment the ground quickly and effectively.

References

[1]. Thrun S, Montemerlo M, Dahlkamp H, et al. Stanley: The Robot That Won the DARPA Grand Challenge[J]. Journal of Field Robotics, 2006, 23(9):661-692.

[2]. Kammel S, Ziegler J, Pitzer B, et al. Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge[J]. Journal of Field Robotics, 2008, 25(9):615–639.

[3]. Moosmann F, Pink O, Stiller C. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion[J]. 2009:215-220.

[4]. Spinello L, Triebel R, Siegwart R. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2008:1823-1829.

[5]. Klasing K, Wollherr D, Buss M. A clustering method for efficient segmentation of 3D laser data[C]// IEEE International Conference on Robotics and Automation. IEEE, 2008:4043-4048.

[6]. Hoffman R, Jain A K. Segmentation and classification of range images. [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2009, PAMI-9(5):608-620.

[7]. Yamauchi H, Lee S, Lee Y, et al. Feature sensitive mesh segmentation with mean shift[C]// International Conference on Shape Modeling and Applications. IEEE, 2005:236-243.

[8]. Teng M, Xiang L, Lu F, et al. Point Cloud Segmentation Based on Spectral Clustering[J]. Journal of Computer-Aided Design & Computer Graphics, 2012, 24(12):1549-1558.

[9]. Zhao J, Li D. Ancient architecture point cloud data segmentation based on modified fuzzy C-means clustering algorithm[C]// International Conference on Earth Observation Data Processing and Analysis. International Society for Optics and Photonics, 2008:72851I-72851I-10.

[10]. Hernandez J, Marcotegui B. Point cloud segmentation towards urban ground modeling[C]// Urban Remote Sensing Event. IEEE, 2013:1-5.

[11]. Himmelsbach M, Hundelshausen F V, Wuensche H J. Fast segmentation of 3D point clouds for ground vehicles[C]// Intelligent Vehicles Symposium. IEEE, 2010:560-565.