2020 International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)

Segmentation Algorithm for 3D LiDAR Point Cloud Based on Region Clustering

Bing Zhou
the College of Information Science and Technology
Beijing University of Chemical Technology
Beijing, China
2018310054@mail.buct.edu.cn

Ran Huang
the College of Information Science and Technology
Beijing University of Chemical Technology
Beijing, China
huangran@mail.buct.edu.cn

2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS) | 978-1-7281-6246-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCSS52145.2020.9336862

***Abstract*—Environmental perception is a key technology in autonomous cars and mobile robotics system. The clustering and segmentation of the point cloud data obtained by sensors is an important step to realize environmental perception. In order to solve the shortcomings of over-segmentation in some current segmentation algorithms, a fast segmentation algorithm for 3D LIDAR cloud points is proposed. The Ground Plane Fitting algorithm is used to segment the ground surface, remove the ground cloud points interference, and combine the spatial Euclidean distance parameter with the angle parameter of the adjacent scan line of the LIDAR to cluster and segment the non-ground cloud points. The experimental result show that compared with some traditional algorithms, this method can effectively reduce the segmentation error rate, improve the accuracy of non-ground target segmentation, and significantly improve the segmentation efficiency.**

***Index Terms*—3D LIDAR, point cloud, clustering, segmentation**

I. Introduction

In the field of mobile robotics and autonomous cars, there are three main ways to realize environmental perception: visual detection, LIDAR detection and multi-sensor fusion detection. Visual detection relies mainly on monocular and binocular cameras. Although they have the advantage of low cost, they are susceptible to interference from external environmental factors(e.g.,light), which limits its application in practical scenarios. In recent years, LIDAR have become popular. LIDAR is a kind of sensor that can obtain environmental information quickly and accurately, compared with visual sensors, LIDAR have the advantages of large detection range, high measurement accuracy, good directionality, less interference by the surrounding environmental factors, etc., which make them widely used in research fields such as automatic cars and mobile robotics. We can accurately identify and track static and dynamic targets in the surrounding environment with the help of LIDAR. In order to achieve this goal, we must first perform accurate data processing and information extraction on a large number of disordered point cloud collections, where a key step is to divide the point cloud information into multiple individuals with actual physical meaning, such as cars, pedestrians,trees, etc. How to accurately realize the clustering and segmentation of each individual has important research significance.

Currently, the research on point cloud segmentation can be roughly divided into several directions, such as model fitting, boundary-based, graph-based, region growth, and attribute-based. The segmentation algorithms based on model fitting use geometric figures to segment the point cloud sets. These algorithms have pure mathematical principles and have a very fast processing speed for simple geometric shapes, even in the face of abnormal values and noise also have good robustness, but they cannot be applied to complex shapes [1], [2], [3]. The RANdom SAmple Consensus (RANSAC) belongs to model fitting algorithm; The boundary-based segmentation algorithm performs segmentation by detecting the boundary of the point cloud data [4], mainly using local features of the point, such as curvature, normal vector, etc. This kind of method can achieve fast segmentation, suitable for scattered and obvious boundaries of the object. However, the effect is not ideal in the face of uneven point cloud density and noise; The graph-based segmentation algorithm is to convert the point cloud data into a graph structure for research [5], [6], [7]. This kind of algorithm can process point cloud data in a complex three-dimensional scene, and has good robustness to noise and point cloud density characteristics, but the complexity of the algorithm is large and not real-time; The segmentation algorithm based on region growth needs to select seed points first, and then perform region growth according to similar feature growth rules [8], [9]. For example, Christoph et al. [10] proposed a segmentation algorithm based on local surface convexity using the principle of region growth. The segmentation algorithm based on region growth has strong robustness, but the segmentation edge has more noise and is prone to over-segmentation; The attribute-based segmentation algorithms often assign each grid face or point cloud data to regions similar to its features by clustering algorithm to obtain segmentation results [11], [12]. This kind of algorithm has strong robustness and accurate segmentation results. However, there is a strong dependence on feature attributes and the algorithm is inefficient. Different clustering algorithms have different advantages and disadvantages and different applicable conditions. Specifically, it can be analyzed in terms of data processing capabilities, algorithm model presets, and model processing capabilities.

In this paper, we propose an unsupervised clustering algorithm, which can achieve more accurate and efficient segmentation of point cloud targets while ensuring the real-time performance of the algorithm. At the same time, it improves the over-segmentation disadvantages existing in some traditional segmentation algorithms. The effectiveness of the algorithm is verified on the KITTI public dataset.

II. METHODOLOGY

The LIDAR scans the environment to obtain point cloud data, and then input them into the system for processing, where each frame of point cloud data is defined as a set:

$$P = \{P_i | P_i = (x_i, y_i, z_i) \in R^3, i = 1, \ldots, n\} \qquad (1)$$

The clustering algorithm first needs to remove the points P_g belonging to the ground plane from the point cloud set P, this step is necessary and it can reduce the number of point clouds that need to be processed, which is beneficial to improve the efficiency and segmentation accuracy of the algorithm. The ground segmentation stage adopts the Ground Plane Fitting algorithm, which can handle the situation where there is a certain slope change. After the ground segmentation is completed, the algorithm processes the non-ground points P_{ng}, and divides the information clusters with actual physical meaning in the point cloud data into a series of independent individuals. Non-ground points segmentation uses the Euclidean distance parameter between the point clouds to complete the clustering by comparing with the set threshold, the distance threshold will be adjusted with the change of the LIDAR scanning range. In order to improve the segmentation accuracy of the algorithm, an angle threshold parameter is added as a basis for judgment. The specific algorithm details are as follows:

A. Ground Segmentation

The data points belonging to the ground plane occupy a significant portion of the point cloud data set. Cloud points that belong to the ground surface removal is a filtering method, which is an effective means to improve the segmentation accuracy of the algorithm. It is easy to identify the cloud points belonging to the ground surface, because these points are basically located on the same plane, and the plane is a simple mathematical model. We can use some algorithms to find a suitable plane model, and then use the height value of the cloud points to complete the ground segmentation, this work is easy to achieve.

Usually in actual scenes, the ground surface is not a perfect plane, there are often slopes, potholes, etc., and as the scanning range of the LIDAR increases, a certain amount of noise will be generated. Therefore, in order to achieve accurate ground segmentation, we cannot simply treat factors such as slope and potholes as non-ground points. Under this restriction, a single plane model is not enough to describe the actual ground surface, so we perform multi-segment fitting on the ground surface. The general idea is to divide the ground surface into N_{segs} sub-planes along the x-axis, and then use the ground plane fitting algorithm for each sub-plane to obtain a plane model similar to the real ground surface.

Algorithm 1: The Ground Plane Fitting algorithm to remove the ground surface

Result:
P_g : ground surface points;
P_{ng} : non-ground target points ;

Initialize:
P : point cloud set
N_{iter} : number of iterations
LPR : the lowest point representative
N_{LPR} : number of the lowest point representative
P_{seeds} : seed points set
H : orthogonal projection distance from points to plane $model$
Th_{seeds} : threshold used to select the seed points set
Th_{dist} : threshold used to determine whether points belong to the ground surface

SelectSeedPointSet:
$P_{sorted} =$ **SortOnHeight**(P);
$LPR.height = Average(P_{sorted}(1 : N_{LPR}))$;
for $k = 1 : |P|$ **do**
  if $p_k.height < LPR.height + Th_{seeds}$ **then**
    $P_{seeds} \leftarrow p_k$;
  end
end
Main Loop:
$P_g = P_{seeds}$;
for $i : N_{iter}$ **do**
  $model =$ **EstimatePlane**(P_g);
  clear(P_g, P_{ng});
  for $j = 1 : |P|$ **do**
    if $H(p_j) < Th_{dist}$ **then**
      $P_g \leftarrow p_j$;
    else
      $P_{ng} \leftarrow p_j$;
    end
  end
end

The first step of the GPF algorithm [13] needs to select an initial seed point set. These seed points are points with a small height (i.e., z-values) in the point cloud set, and are mainly used to establish an initial plane model describing the ground surface. In order to select the seed point set, we select N lowest height points in the point cloud set and calculate their average value, which is defined as the lowest point representative (LPR).The LPR reduces the influence of measurement noise on the plane fitting. We set a height threshold Th_{seeds}. When the height of some points in the point cloud P is lower than the sum of Th_{seeds} and the LPR, these points are regarded as seed points, and these seed points constitute a set of seed points.

After that, a plane can be determined according to the set of seed points, and we use a simple linear model for plane

978-1-7281-6246-1/20/$31.00 ©2020 IEEE

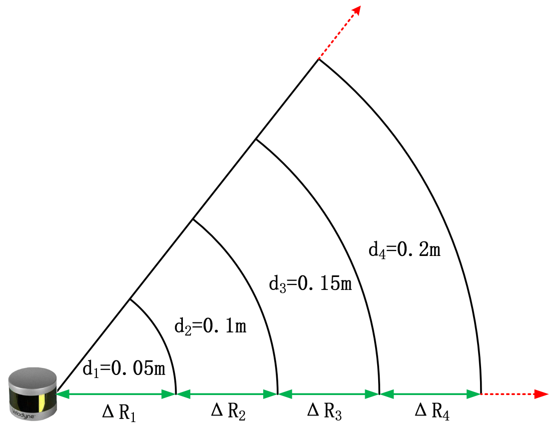

Fig. 1. The distance threshold d increases as the scanning range increases. As the scan range of the LIDAR sensor increases, the distance threshold d will increase, which to some extent can solve the over-segmentation problem caused by the cloud points becoming sparse.

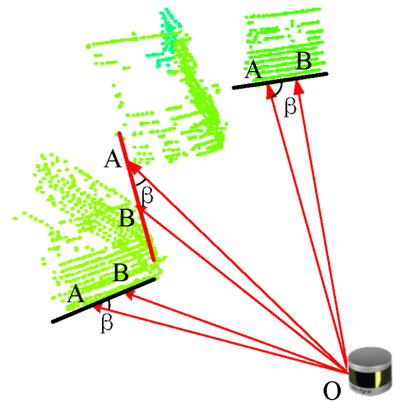

Fig. 2. A top view on the pedestrians from the example scene. There are two situations: When $\beta < \alpha$, we consider the two points A and B belong to two different objects and the connection between the two points is represented by red lines; when $\beta > \alpha$, we consider the two points A and B belong to the same object and are represented by black lines.

model estimation:

$$ax + by + cz + d = 0 \tag{2}$$

that is

$$\mathbf{n}^T\mathbf{x} = -d \tag{3}$$

where $\mathbf{n} = [a\ b\ c]^T$, $\mathbf{x} = [x\ y\ z]^T$, $\mathbf{n}$ can be solved by the covariance matrix $C \in R^{3\times3}$ of the initial seed point set. The set of seed points $S \in R^3$ is used as the initial point set, and its covariance matrix is:

$$C = \sum_{i=1:|S|}(s_i - \hat{s})(s_i - \hat{s})^T \tag{4}$$

where $\hat{s}$ is the average value of all points in the set of seed points. The covariance matrix C describes the distribution of the point set, and its three singular vectors can be calculated by singular value decomposition (SVD). Since it is a plane model, the normal vector $\vec{n}$ perpendicular to the plane represents the direction with the smallest variance.

After acquiring the plane model, the orthogonal projection distance of each point in the set of point cloud to the plane is calculated, and compare the projection distance with the set distance threshold Th_{dist}. If the projection height is less than the Th_{dist}, the point is considered to belong to the ground surface, otherwise it belongs to non-ground targets.

B. Non-ground Target Segmentation

After the point cloud set P is processed by ground segmentation, the next step is to cluster the non-ground points P_{ng}, which can be used for target detection, tracking, etc. To achieve accurate clustering of independent objects is a challenging task. The traditional Euclidean clustering algorithms use Euclidean distance parameters to achieve the purpose of clustering, usually set the distance threshold to a fixed parameter, which will cause many disadvantages, because as the LIDAR scan range increases, the cloud points acquired by the sensor will change from dense to sparse. If the set distance threshold is too small or too large, over-segmentation and under-segmentation will often occur, i.e., splitting an object into multiple clusters and clustering multiple clusters into a single target. To solve this problem, we propose an adaptive method to determine the distance threshold d according to the change of the LIDAR scanning range (Fig. 1), which can be expressed as:

$$d = 2r\tan\frac{\theta}{2} \tag{5}$$

where r is the scan range of the 3D LIDAR, θ is the fixed vertical angular resolution and $\theta = 0.4°$, determined by the LIDAR, d is the vertical distance between two adjacent laser scans. According to the expression, as the scan range of the LIDAR sensor increases, although the cloud points will become sparse, the distance threshold d will increase accordingly [14], which to some extent can solve the over-segmentation problem caused by the cloud points becoming sparse.

In three-dimensional space, compared with vision sensors, LIDAR sensors have a larger scan range and higher accuracy, thus requiring more point cloud data to be processed, which greatly reduces the clustering speed of the algorithm and limiting its application in actual scenarios. In order to improve the efficiency of the algorithm, we divide the space into nested sector regions centered on the LIDAR sensor, apply different distance thresholds in different sector regions, use KD-tree algorithm for neighborhood search, and set the change value of the distance threshold between rings to 0.05m, that is

$$\begin{aligned} d_{i+1} &= d_i + \Delta d, \\ \Delta d &= 0.05m \end{aligned} \tag{6}$$

obviously, according to Eq. 5 and Eq. 6, the ring with width Δr can be calculated to be between 7m-7.5m, we take an

Algorithm 2: Pseudocode of non-ground targets segmentation

Result: C : the list of point clusters

P : point cloud set
Q : a queue of the points that need to be checked
α : the angle threshold
β : the angle between the connecting line of two points in the point cloud dataset and the longer scan line
R : the distance
Th_{dist} : the threshold distance

create a Kd-tree representation for the point cloud dataset P
$Q = \phi$;
for *each* $p_i \in P$ **do**
  $Q \leftarrow p_i$;
  for $p_i \in Q$ **do**
    $r = R(p_i, LIDAR)$; //compute Th_{dist} by r and Eq. 5
    for *find* p_k^i *for point* p_i *in a sphere by Kd-tree* **do**
      $\beta = \beta(p_i, p_k^i)$;
      if $R(p_i, p_k^i) < Th_{dist}$ *and* $\beta > \alpha$ **then**
        p_i, p_k^i belong to the same object;
        $P_k^i \leftarrow p_k^i$;
      end
      for *each* $p_k^i \in P_k^i$ **do**
        if p_k^i *has not been processed* **then**
          $Q \leftarrow p_k^i$;
        end
      end
    end
    if *all points in* Q *has been processed* **then**
      $C \leftarrow Q$;
      $Q = \phi$;
    end
  end
end
the algorithm terminates when each point $p_i \in P$ have been processed and are now part of the list of point clusters C

integer for the ring width $\Delta R = \lfloor \Delta r \rfloor$ with:

$$\Delta r = \frac{\Delta d}{2} \cot \frac{\theta}{2} \tag{7}$$

When LIDAR perceives environmental information, a large angle is formed between the reflective point connections of adjacent scan lines on the same object surface and longer scanlines, while a small angle is formed between the reflective point connections of adjacent scan lines on different object surfaces and longer scan lines. According to this characteristic, Bogoslavskyi et al. [15] propose an algorithm to achieve fast segmentation of different targets. In this paper, we use the angle threshold judgment method as a supplement to the algorithm, which can better solve the problem of over-segmentation of adjacent targets (Fig. 2). The point O represents the LIDAR, the points A and B represent the nearby reflection point of the scanned object by the LIDAR. OA and OB are compared to pick up the long scan line OA, according to Eq. 8, the angle β between OA and AB can be calculated:

$$\beta = \arccos \frac{||OA||^2 + ||AB||^2 - ||OB||^2}{2 * ||OA|| * ||AB||} \tag{8}$$

set the angle threshold α, if the value of β is greater than α, then the points A and B belong to the same object, otherwise they belong to different objects.In this paper,the algorithm uses the spatial Euclidean distance parameter as the main indicator when clustering multiple targets, and the angle threshold parameter as a supplement to complete the fast segmentation of multi-target individuals.

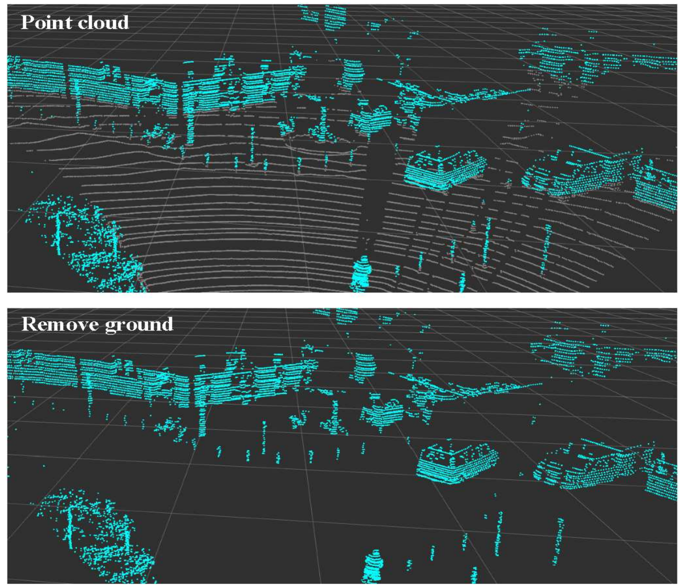

Fig. 3. The ground surface is often not a perfect plane in actual scenes. In order to deal with slopes, potholes, etc,. we use the Ground Plane Fitting algorithm when segmenting the ground surface.

III. EXPERIMENTAL EVALUATION

The purpose of this work is to provide a fast and effective segmentation method for 3D point cloud data, which can assist mobile robotics or autonomous vehicles platforms to understand environmental information. In order to verify the segmentation accuracy and real-time performance of the algorithm, the experimental data uses the KITTI public dataset [16] for testing, and we select 200 scans from the KITTI raw dataset to compare our approach and Euclidean Cluster Extraction (PCL). The sensor use Velodyne HDL-64E 3D LIDAR with a horizontal field of view of 360°, a vertical field of view from $+2°$ to $-24.8°$, and an operating frequency of 10Hz. We use a laptop with i5-9300H 2.4GHz CPU for experiments.

Ground plane segmentation uses the GPF algorithm, which can deal with the existence of a certain slope. Some of

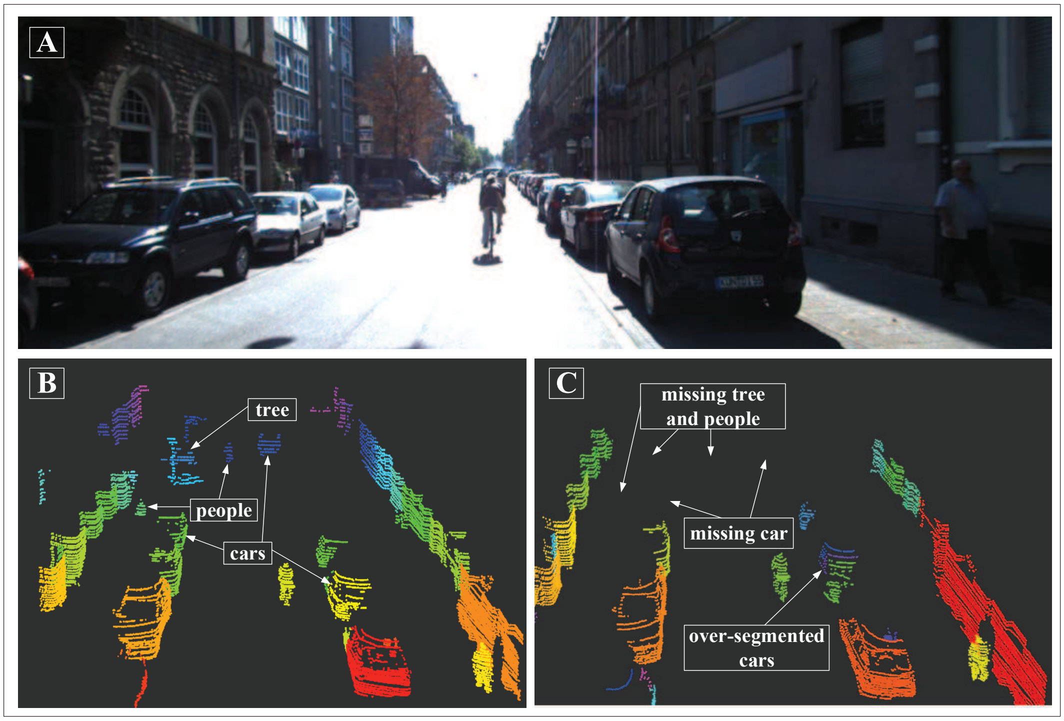

Fig. 4. A : The raw image of an outdoor scene from the KITTI raw dataset; B : Our approach : when facing the outdoor scene, our approach can deal with sparse cloud points at a distance and produce a good segmentation result; C : Euclidean Cluster Extraction : ECE algorithm segment the outdoor scene. There are some objects that are situated further from the LIDAR missing, such as cars, people, trees, and one car is divided into two objects.

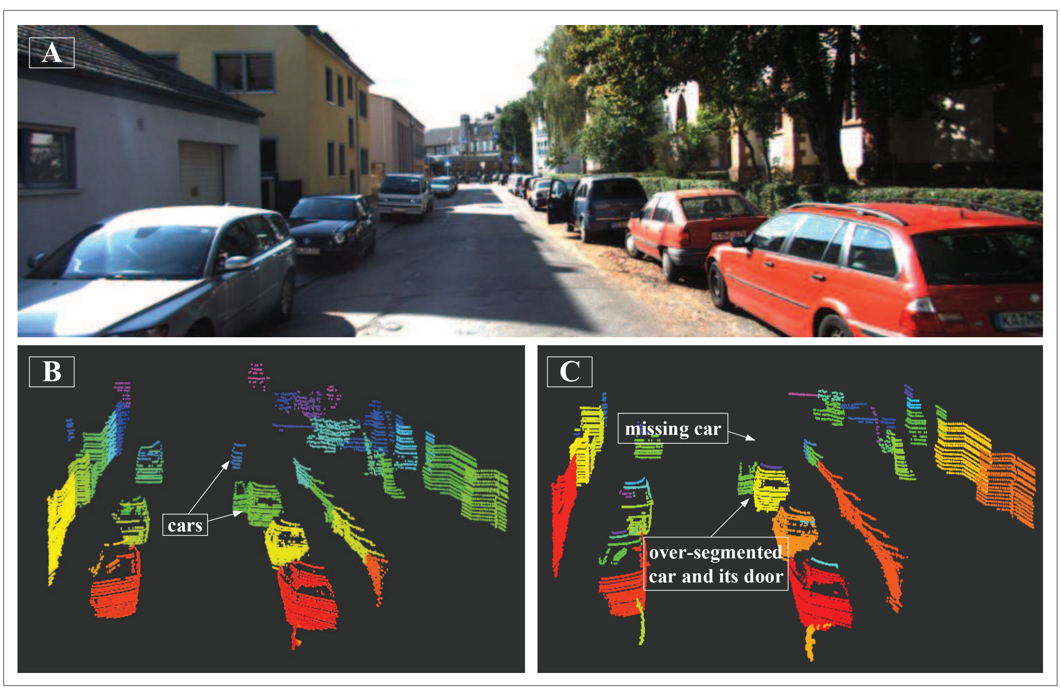

Fig. 5. A : The raw image of an outdoor scene from the KITTI raw dataset; B : Our approach : our approach segment the outdoor scene without over-segmenting the close objects; C : Euclidean Cluster Extraction : ECE algorithm segment the outdoor scene. There is an over-segmentation situation that one car and its door are divided into two separate objects, another car further away from the sensor is missing.

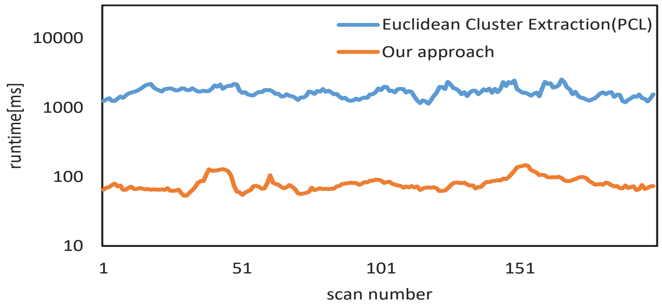

Fig. 6. We select 200 scans from a 64-beam Velodyne dataset (KITTI), and compare our approach and Euclidean Cluster Extraction (PCL) in the runtime.

the relevant parameters are set: $N_{segs} = 2$, $N_{iter} = 10$, $N_{LPR} = 120$, $Th_{seeds} = 0.2m$, $Th_{dist} = 0.3m$. The ground segmentation effect is shown in Fig. 3.

For the non-ground targets segmentation algorithm, we combine the two parameters of spatial Euclidean distance threshold and angle threshold to cluster the cloud points. For the angle threshold, we set $\alpha = 10°$. Our algorithm has obvious advantages in processing speed over Euclidean Cluster Extraction (ECE) algorithm, as shown in Fig. 6. In terms of algorithm segmentation accuracy, compare our algorithm with ECE algorithm, our algorithm has higher accuracy and can effectively overcome the over-segmentation in some other clustering algorithms (Fig. 4 and Fig. 5). And when facing sparse point clouds, our algorithm can also obtain good segmentation results.

IV. CONCLUSIONS AND FUTURE WORK

This paper proposes a fast and effective point cloud segmentation algorithm, including two links: ground segmentation and non-ground target segmentation. Considering that the ground surface has a certain slope, the ground segmentation uses the Ground Plane Fitting algorithm. In order to overcome the under-segmentation defects in some segmentation algorithms, in the segmentation of non-ground targets, we combine the spatial Euclidean distance threshold and the angle threshold to determine whether the point belongs to the non-ground targets or not. We have conducted experiments using the KITTI public dataset, and the results show that the algorithm in this paper is faster than some algorithms in execution speed, which makes it very suitable for real-time operation of data collected by LIDAR, and also has obvious advantages in segmentation effects, even facing sparse cloud points.

Segmentation of cloud points is the first step to realize environmental perception. Our future work is to realize the recognition of dynamic obstacles and static obstacles, and to track dynamic obstacles.

REFERENCES

[1] D. Chen, L. Zhang, P. T. Mathiopoulos and X. Huang,“A Methodology for Automated Segmentation and Reconstruction of Urban 3-D Buildings from ALS Point Clouds,” in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 10, pp. 4199-4217, Oct. 2014.

[2] M. Awadallah, L. Abbott and S. Ghannam, “Segmentation of sparse noisy point clouds using active contour models,” 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 6061-6065.

[3] D.Yan, W. Wang and Y. Liu, “Variational mesh segmentation via quadric surface fitting,” Computer-aided Design, 2012, 44(11): 1072-1082.

[4] E. Castillo, J. Liang and H. Zhao, “Point Cloud Segmentation and Denoising via Constrained Nonlinear Least Squares Normal Estimates,” In: Breuβ M., Bruckstein A., Maragos P. (eds) Innovations for Shape Analysis. Mathematics and Visualization. Springer, Berlin, Heidelberg, 2013.

[5] J. Strom, A. Richardson and E. Olson, “Graph-based segmentation for colored 3D laser point clouds,” 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 2131-2136.

[6] J. Yang, Z. Gan, K. Li and C. Hou, “Graph-Based Segmentation for RGB-D Data Using 3-D Geometry Enhanced Superpixels,” in IEEE Transactions on Cybernetics, vol. 45, no. 5, pp. 927-940, May 2015.

[7] J. R. Schoenberg, A. Nathan and M. Campbell, “Segmentation of dense range information in complex urban scenes,” 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 2033-2038.

[8] A. Nurunnabi, D. Belton and G. West, “Robust Segmentation for Large Volumes of Laser Scanning Three-Dimensional Point Cloud Data,” in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 8, pp. 4790-4805, Aug. 2016.

[9] S. C. Stein, M. Schoeler, J. Papon and F. Wörgötter, “Object Partitioning Using Local Convexity,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 304-311.

[10] Y. Xu, W. Yao, L. Hoegner and U. Stilla, “Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing,” Remote Sensing Letters, 2017, 8(11): 1062-1071.

[11] J. M. Biosca, J. L. Lerma, “Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods,” Isprs Journal of Photogrammetry and Remote Sensing, 2008, 63(1): 84-98.

[12] D. Holz, S. J. Holzer, R. B. Rusu and S. Behnke, “Real-time plane segmentation using RGB-D cameras,” robot soccer world cup, 2012: 306-317.

[13] D. Zermas, I. Izzat and N. Papanikolopoulos, “Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications,” 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 5067-5073.

[14] Z. Yan, T. Duckett and N. Bellotto, “Online learning for human classification in 3D LiDAR-based tracking,” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 864-871.

[15] I. Bogoslavskyi and C. Stachniss, “Fast range image-based segmentation of sparse 3D laser scans for online operation,” 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016, pp. 163-169.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” International Journal of Robotics Research, 2013, 32(11): 1231-1237.