Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Unsupervised Segmentation Of Urban 3D Point Cloud Based On Lidar-image

Xia Yuan, *Member, IEEE*, Yangyukun Mao, Chunxia Zhao

School of Computer Science and Engineering
Nanjing University of Science and Technology
Nanjing, Jiangsu, China

yuanxia@njust.edu.cn

***Abstract* - As one of the most vital fields in 3D scene understanding, segmentation of lidar point cloud has verity of applications such as unmanned vehicle. In order to deal with the problem of limited local computing resources, this paper proposes a two-step unsupervised learning-free method to segment sparse 3D point cloud with low computational demands. The 3D point cloud data is projected into a lidar-specific 2D coordinate system called lidar-image. For unmanned vehicles, road is a special kind of object, so the algorithm divides road firstly by using improved lidar-histogram. The algorithm further examines the geometric connection between adjacent points in 3D space, thus dividing non-road points into independent subsets. Experiments on KITTI dataset validate the proposed algorithm outperforms similar methods.**

Index Terms* – *point cloud, segmentation, lidar-image, lidarhistogram.

I. Introduction

With the ability of providing long-range and high-accuracy 3D point cloud data, lidar is an important sensor for a Unmanned ground vehicle (UGV) to perceive the surrounding environment. It is still a big challenge to understand 3D point cloud.

City is a typical complex environment for a UGV which contains a large number of different types of objects. Segmentation is an effective way to deal with this kind of cluttered point cloud. Good data segmentation is of great significance to object recognition and environment understanding. Although deep learning-based end to end model is popular now, there are still many problems in the practical application of these models due to the lack of 3D point cloud training datasets for semantic segmentation and object detection. In this case, unsupervised segmentation model is very useful to deal with 3D point cloud of real scene.

Road and objects on the road are very important for a UGV to navigate itself in city. We focus to segment these objects with an unsupervised learning-free method. 3D points are projected into 2D as lidar-image to reduce the complexity of the algorithm in neighborhood computing. Practice has shown that splitting the road first helps to better segment other objects in the city's driving scene. We propose a two-step segmentation model in this paper.

The main contributions of this paper lie in: (i) we proposed a two-step unsupervised learning-free model which can work on the platform with limited local computing resources; (ii) an improved lidar-histogram based ground segmentation algorithm is proposed to fast segment road in urban environment; (iii) this paper proposes a connected neighborhood segmentation algorithm to segment non-road area, and experiments valid that it works well to segment objects on the road like pedestrians, cyclists, cars, vans, traffic signs, etc..

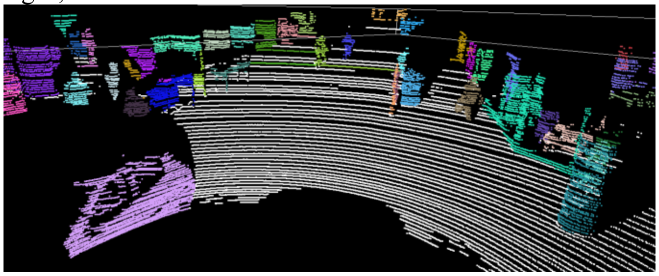


Fig.1 Segmentation results of the proposed method. White points represent the ground, and points in other colors represent different sub-clusters.

The rest of this paper is organized as follows. In the following sections, section II introduced some related work. The specific description of the proposed algorithm is in section III. Section IV shows experiment results. Finally, the conclusion are presented in section V.

II. Related Work

A lot of research work has been done in road segmentation and object detection based on 3D point cloud. Shinzato et al. [2] proposed a sensor fusion-based method using a minimum number of parameters to detect road. Their approach was based on the spatial-relationship on perspective images provided by a single camera and a 3D Lidar. Also, Hu [3] used RANSAC to estimate ground points directly. They project the ground points onto a 2D image and use the corresponding pixels to learn a Gaussian model. Chen et al. [4] proposed a histogram-based algorithm to extract road which is fast and effective. We have improved this algorithm and formed the ground segmentation algorithm in this paper.

For the segmentation of non-road points, many object detection algorithms for 3D point cloud have been proposed. Douillard et al. [5, 6] proposed a feature-based approach. This kind of methods define geometric characteristic operators and

* This work is supported by National Natural Science Foundation of China under Grant #61603184 and #61773210.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

implement the segmentation on 3D points directly, which are accurate but time consuming. While Korchev et al. [7] focus on projecting the lidar points onto 2D occupied grid map and then segmenting the grid cells. This method is fast but its accuracy is relatively low because the discretization of the grid may not be suitable for the current environment, different objects may be grouped into one cluster. [8] and [9] used 3D points to create a lidar-image and use a single parameter for calculation.

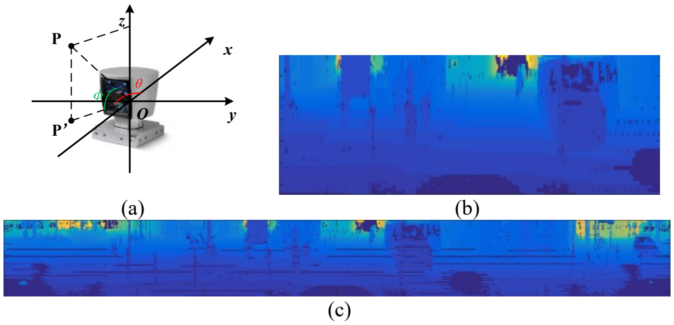
CNN based models are very popular in recent years. Some of them were based on color images and lidar projection images [12-14], while some of them directly convoluted in 3D space [15. The generalization ability of these CNN based algorithms are still needs to be verified in practice, and they require high computational resources, especially when dealing with large-scale outdoor scenes.

III. Image-based Unsupervised Segmentation

The algorithm proposed in this paper includes three steps: First step, projecting a 3D point cloud into a 2D to generate lidar-image. Secondly, improve the lidar-histogram algorithm and apply it to road segmentation. The last step is to segment non-road points by using connected neighborhood segmentation algorithm.

A. Lidar-image

We use Velodyne HDL-64E S3 as an example to introduce how to build lidar-image. Each lidar point system contains three parameters in spherical coordinate, which are rotation angle θ, vertical angle φ and measurement distance d. The relation of spherical coordinate system (θ, φ, d) and Cartesian coordinate system (x, y, z) is shown in Fig.2 (a). Since there are 64 laser detectors of a HDL-64E and each laser detector provides 2000 points per 360° of rotation, we define an image with resolution of 64×2000 pixels and project a lidar point into a pixel by index their coordinate (θ, φ) as column and row in the image.



(a)

(b)

(c)

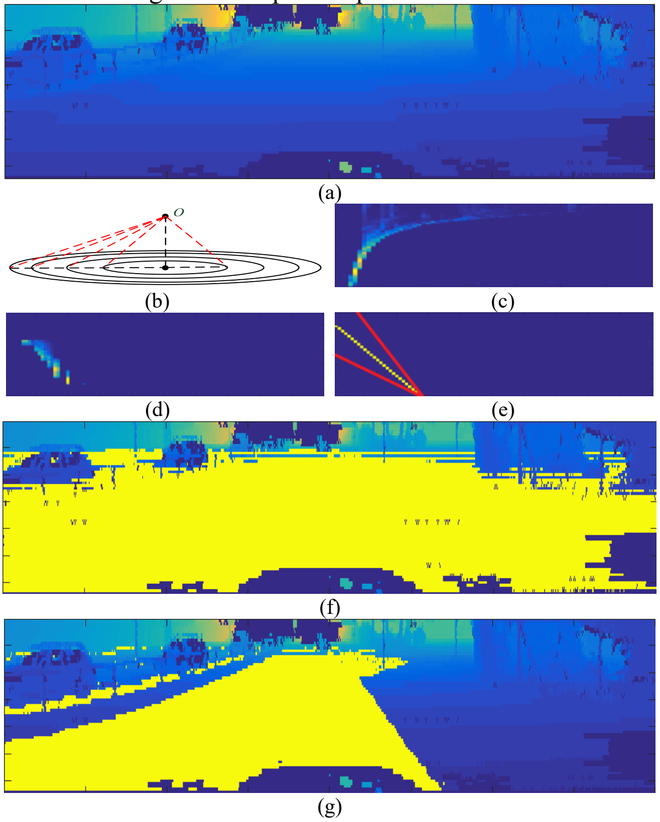
Fig.2 The coordinate and lidar-image (a) the coordinate system, (b) lidar-image of in front of the car, cutting and bilinear interpolating from (c), (c) lidar-image of 360°.

For example, we define the point (0°, 2.0°) as the pixel [0, 0] and the point (0.18°, 2.0°) as [1, 0] in the image. Thus all points are arranged in pixels to form an image what we call lidar-image and the value of one pixel in it is the corresponding measurement distance d. Fig.2 (c) shows a lidar-image with pseudo-color. There are some missing pixels appear as horizontal stripes on Fig.2 (c), and we use bilinear interpolation algorithm [10] to fill them.

Since a UGV pays more attention to the area in front of it, we select the 601th column to the 1400th column in the lidar-image which cover 144° in front of the vehicle and reform as Fig.2 (b) to be segmented in the next .

B. Improved lidar-histogram based ground segmentation

The algorithm will find ground pixels firstly. Without loss of generality, it is assumed that the ground area is flat in urban environments like Fig.3 (b). So, value of pixels which belong to the ground on the same row in a lidar-image should be about the same. Different with [4], we use d of lidar points in spherical coordinate to be the value of pixels on liar-image. Since the algorithm only use lidar data in front of the vehicle, it is suppose that there are lot of pixels in each row of the lidar-image belonging to the ground, especially in the lower part of the image which represent points close to the vehicle.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig.3 Ground segmentation. (a) lidar-image, (b) ground hypothesis, (c) lidar-histogram of Fig.2(b), (d) transformed lidar-histogram, (e) result of RANSAC, (f) initial ground, (g) refined ground.

We set the quantization interval of the histogram as 1m, and the total range is set to be a integer interval of [1, 70] as laser points that 70m away are too sparse to be understood. For each row of the lidar-image, the number of quantized pixels' value are counted to form a 2D histogram as Fig.3 (c). The size of Fig.3 (c) is 64×70. Ground pixels will accumulate in each row in lidar-histogram based on the previous assumptions and we did observe this phenomenon in Fig.3 (c).

Highlight pixels in Fig.3 (c) are not linearly distributed, so we transform Fig.3 (c) to Fig.3 (d) to make ground points more linearly distributed in lidar-histogram. We take reciprocal value of pixels in Fig.3 (c) and multiply them by a weight w, then round up the result. The rounding rule is rounded down if the fractional part is less than 0.2, and rounded up if it is not less than 0.2. We take w=100 in this paper and get a transformed lidar-histogram as Fig.3 (d) with size 64×100.

Ground points are almost linearly distributed in Fig.3 (d), and we use RANSAC algorithm to obtain a linear model. Suppose this model is y=kx+b, where x, y are number of column and row in transformed lidar-histogram. Then we set up α and β to be the threshold because of the lidar measurement error and tiny uneven of the ground and get the range of ground [x$_1$, x$_2$] as (1)

$$\begin{aligned} x_1 &= \frac{y-b}{k} + \frac{\alpha}{y}, \alpha > 0 \\ x_2 &= \frac{y-b}{k} + \frac{\beta}{y}, \beta < 0 \end{aligned} \tag{1}$$

where we divide α and β by y because with the extension of the plane, the laser beams become sparse and far away from each other. Equation (1) means that if a pixel in a row that y corresponds to have the value in range from x_1 to x_2, this pixel can be put into the set of ground as shown in Fig.3 (e). Fig.3 (f) shows the segmentation result of ground points and it looks very bad intuitively. The key problem is that the ground we are talking about in this paper is actually means the road in urban environment for a UGV. With the above method, road isolation zones such as road curb and grassland can easily be segmented into ground. So we need to refine Fig.3 (f) to get Fig.3 (g) using refined road scanning method.

The refined road scanning method we proposed is simple and efficient. When a UGV moving on the road, we reasonably believe that the middle point at the bottom of the lidar-image should belong to road. Starting with this point, the algorithm scans left and right points at the same row of lidar-image respectively.

Taking the left scan as an example. The depth of the starting point is taken as the reference value of road depth, and the difference between the depth of the left point and the reference value is compared one by one. If the depth difference is not over a threshold, the scanned point is considered to belong to the road, otherwise the scanned point does not belong to the road. For each k points scanned, the depth of the last point which determined to belong to the road is taken as the new reference value of road depth. The right scan process is the same as the left scan process.

After the bottom row of the lidar-image scanning is completed, the scan begins to go up row by row. Each row's scanning start point is depend on the last row' starting point. Suppose the last row' starting point located in the i-th row and j-th column is $p(i, j)$. Comparing the depth difference between point $p(i, j)$ and p(i-1, j-m) to p(i-1, j+m) respectively. The point whose depth difference is the minimum and does not exceed the threshold is taken as the starting point in the (i-1)-th row. In order to adapt to the characteristics of lidar, the depth threshold is increased Δd by scanning after one row upward of the lidar-image.

The refined result of ground segmentation is shown in Fig.3 (g). On the one hand, the result is much better than Fig.3 (f). On the other hand, besides the current direction of the road, this algorithm has the ability to segment and get the opposite lane, which is an advantage of the proposed algorithm.

Taking the left scan as an example. The depth of the starting point is taken as the reference value of road depth, and the difference between the depth of the left point and the reference value is compared one by one. If the depth difference is not over a threshold, the scanned point is considered to belong to the road, otherwise the scanned point does not belong to the road. For each k points scanned, the depth of the last point which determined to belong to the road is taken as the new reference value of road depth. The right scan process is the same as the left scan process.

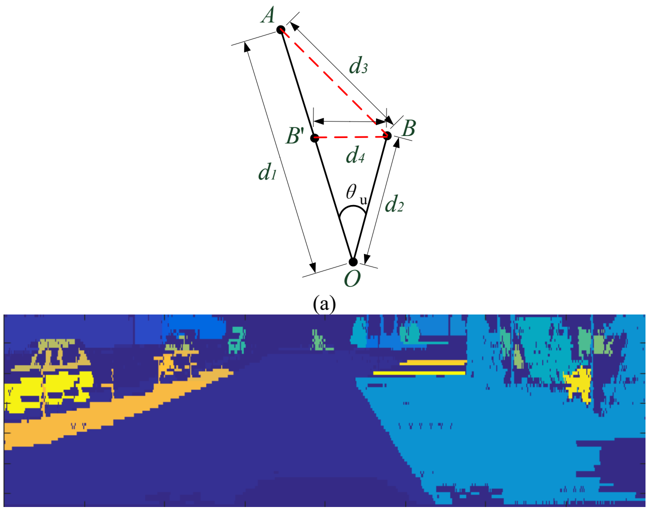
After the bottom row of the lidar-image scanning is completed, the scan begins to go up row by row. Each row's scanning start point is depend on the last row' starting point. Suppose the last row' starting point located in the i-th row and j-th column is $p(i, j)$. Comparing the depth difference between point $p(i, j)$ and p(i-1, j-m) to p(i-1, j+m) respectively. The point whose depth difference is the minimum and does not exceed the threshold is taken as the starting point in the (i-1)-th row. In order to adapt to the characteristics of lidar, the depth threshold is increased Δd by scanning after one row upward of the lidar-image.

The refined result of ground segmentation is shown in Fig.3 (g). On the one hand, the result is much better than Fig.3 (f). On the other hand, besides the current direction of the road, this algorithm has the ability to segment and get the opposite lane, which is an advantage of the proposed algorithm.

A. Connected neighborhood segmentation algorithm

For non-road points, we proposed a method called connected neighborhood segmentation algorithm. The main idea is to calculate whether two adjacent pixels in lidar-image belong to one object or not according to their geometric relationship.

Suppose A and B are two points formed by consecutive laser beams OA and OB of one laser detector respectively in Fig.4 (a). If OA and OB are jetted on a same object, the length of them should be very close. Without loss of generality, we assume that the length of OA is not less than OB. We define B' on OA that satisfied ||OB'||=||OB||. If A and B lie on the same object, the distance between points A and B which is defined as d_3 need to be close to the distance between B and B' defined as d_4. In practice, we use a ratio r between d_3 and d_4 as the basis for judging whether A and B are on the same object. The ratio r is defined by (2).



(a)

(b)

Fig.4 Non-road area segmentation. (a) connected neighborhood segmentation. (b) segmentation result of Fig.2(c).

$$r = \frac{d_3}{d_4} = \frac{\sqrt{d_1^{\,2} + d_2^{\,2} - 2d_1 d_2 \cos\theta_u}}{\sqrt{2d_2^{\,2}(1-\cos\theta_u)}} \tag{2}$$

Where θ_u is the scanning angular resolution which is provided by the setup of the lidar. We use this method to judge whether two adjacent points (pixels) lie on the same object or not. Adjacent we mean here refers to three direction on lidar-image which are horizontal, vertical and diagonal. In other words, we will examine whether a point on a Lidar-image is on the same object with its 8-neighbor respectively. For different direction we use different θ_u as a lidar's horizontal and vertical scanning angular resolution are usually different, so θ_u will have three values defined by the setup of the lidar.

We define a threshold r0 that represents the maximum value for connecting these two points. If the r of two adjoining points is greater than r_0, we consider these two points belong to different objects. Otherwise, we think these two points belong to the same object.

Alg.1 describes the connected neighborhood segmentation algorithm. We set a tag matrix M and initialize the cluster label Num (line 2-3). The beginning of traversal is from the top left corner of lidar-image and then the scanning pixel pass through every pixel from left to right, top to bottom (line 4-5).

Algorithm 1 Connected neighborhood segmentation algorithm

```
1:  procedure Traversal Process
2:      $M \leftarrow zeros(I_{\text{rows}} \times I_{\text{cols}})$
3:      Num ← 1, $I \leftarrow$ lidar-image
4:      for $i = 1 \ldots I_{\text{rows}}$ do
5:          for j=1…$I_{\text{cols}}$ do
6:              if $M(i,j) = 0$ then
7:                  queue.push($\{(i,j)\}$);
8:                  while queue is not empty do
9:                      $M(i,j) \leftarrow$ Num;
10:                     $\{i_{cen}, j_{cen}\} \leftarrow$ queue.top();
11:                     for $\{i_n, j_n\} \in$ Neighborhood$\{i_{cen}, j_{cen}\}$ do
12:                         $d_1 \leftarrow \max(I(i_{cen}, j_{cen}), I(i_n, j_n))$;
13:                         $d_2 \leftarrow \min(I(i_{cen}, j_{cen}), I(i_n, j_n))$;
14:                         if $\frac{\sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos\theta_u}}{\sqrt{2d_2^2(1-\cos\theta_u)}} \le r_0$ then
15:                             queue.push($\{i_n, j_n\}$);
16:                     queue.pop();
17:             Num ← Num + 1;
```

When the pixel is not marked (line 6), we need to take this pixel as the center for a breadth first search (BFS) in order to find the pixels which lie on the same object as the center pixel in the neighborhood. To realize BFS, we push the neighboring pixels that match the condition of ratio into a queue and use the head of the queue to be the new center to search for more pixels that can be segmented into one object (line 7-line 15). Each pixel in the queue is marked by parameter Num (line 9) and the label number of cluster won't change until the queue is empty to make sure that the pixels in this queue have the same mark. The pixels which have been the center and finished searching will be dequeued (line 16). And when the queue is empty, the BFS on this initial pixel is over and the label number of cluster changes (line 17). After that, the scanning pixel will keep moving until the next pixel that has no mark is encountered.

The execution time of the whole traversal process is very short because we just need to visit each pixel twice in the worst case and the time complexity is O(n) (n is the number of pixels in a lidar-image). Fig.4 (b) is the final segmentation result and we can see that pedestrians on the road, trees and walls are all segmented precisely.

IV. Experiments

We use Kitti-Road and Kitti-Tracking dataset [11] to validate the proposed method. All comparative experiments are executed on a computer with an i7-8700K 3.70 GHz CPU and 16G RAM. We implemented algorithms in [4] and [8] with MATLAB R2014b.

As mentioned earlier in this paper, we cut the horizontal range of $\theta \in$ [108°, 252°] which is 144° ahead of vehicle in the lidar-image. The values of the parameter are $\alpha = 20$, $\beta = -16$, $r_0 = 2.8$. The experiments are divided into two parts, the ground segmentation and non-road segmentation.

A. Ground segmentation result on Kitti-Road dataset

The Kitti-Road dataset has three different categories including Urban Marked (UM), Urban Multiple Marked (UMM), and Urban Unmarked (UU). All categories have ground-truth in their training dataset. We project the marked point clouds onto the ground-truth image based on calibration parameters and calculate the classical pixel-based metric

Precision, Recall and Max F1-measure (MaxF) to valid algorithms. MaxF is calculated by using (3), and β=1.

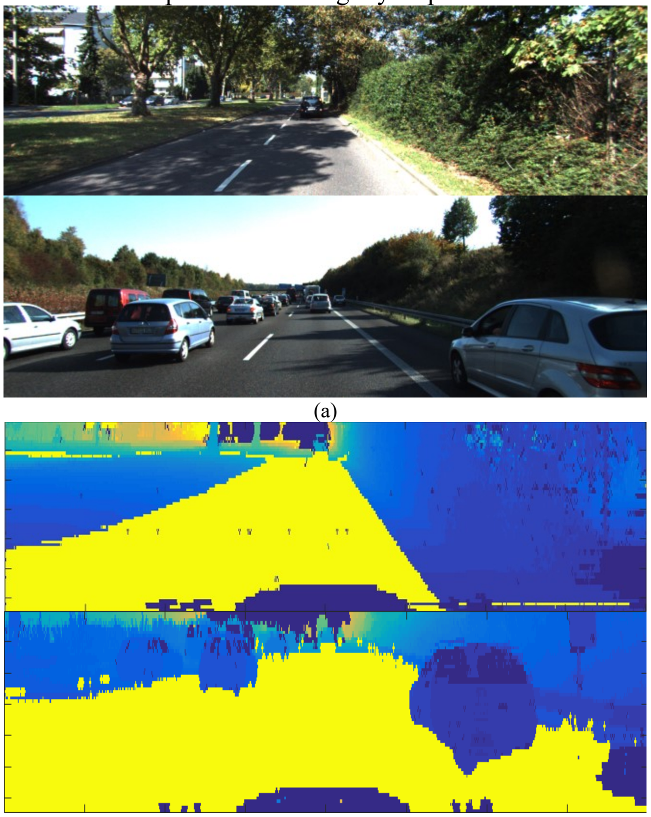
$$F_{\max} = \arg\max[(1+\beta^2)\times\frac{\text{PRE}\times\text{REC}}{\beta^2\times(\text{PRE}+\text{REC})}] \tag{3}$$

Table I shows quantitative experimental results on both methods. Our method got 3%-4% higher than [4] at recall on UM, UMM and UU. The precision of our method is about 6.7% higher than [4] on UM, and 0.6% lower on UMM and UU. And the MaxF of our method is better than [4] according to recall and precision. The runtime of the proposed algorithm has a slight advantage as shown in Tab.I.

TABLE I. Comparison Results On Kitti-road

Category	Method	MaxF	PRE	REC	Runtime
UM	Chen[4]	89.87%	91.28%	88.49%	100ms
	Ours	**93.96%**	**91.95%**	**96.45%**	95ms
UMM	Chen[4]	93.32%	**95.39%**	91.34%	100ms
	Ours	**95.86%**	94.77%	**96.98%**	95ms
UU	Chen[4]	86.55%	**90.71%**	82.75%	100ms
	Ours	**89.89%**	90.09%	**89.70%**	95ms

Fig.5 (a) is the color image of a scene. Fig.5 (b) is the result of our method and it looks good. Curb and off-road area are successfully segmented as non-road which is exactly what we hope to achieve. Fig.5 (c) is the ground segmentation result of [4]. We can find that ground shape is incomplete. We tried to adjust the parameters of the algorithm in [4], but we did not get good results. This situation may be due to the road in the scene is not complete flat and slightly slope.


(a)

(b)

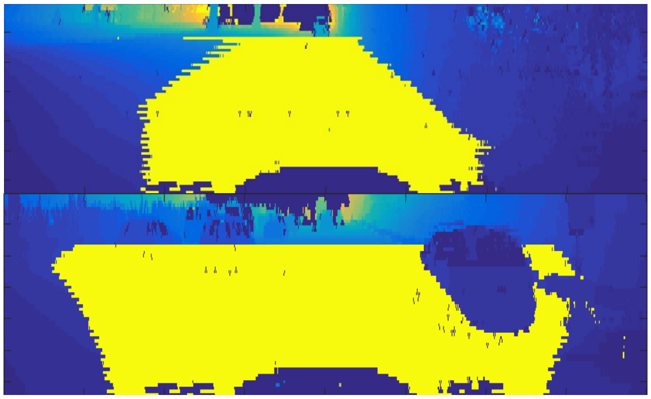

(c)

Fig.5 The results of two methods of ground segmentation. (a) color image of two scenes. (b) result of ground segmentation by the proposed method. (d) result of ground segmentation by algorithm in [4].

From Fig.5 (b) and Fig.3 (g) we can find that the greatest advantage of the proposed ground segmentation algorithm is that it can detect not only the forward lane of the vehicle, but also the reverse Lane across the isolation zone.

We found Kitti-Tracking dataset is good to valid our non-road segmentation algorithm, though we never do object recognition of sub-clusters. Labels of the sub-clusters in the flowing experiments are given by the training data of Kitti-Tracking dataset. The Kitti-Tracking dataset has 21 groups of training dataset. It provides 8 different object labels including pedestrian, cyclist, van, car, etc.. We set different colors for different objects as shown in Fig.6 and Fig.7.

Tab.II shows the quantitative experimental results of non-road segmentation by comparing with the algorithm proposed in [8]. Considering the fairness of algorithm comparison, we use matlab to re-implement the algorithm in [8] and only compare results of non-road segmentation part. Since Kitti-Tracking dataset has no ground truth of pixel-level, we use 2D box of objects to represent the ground truth. We first segment corresponding 3D points, and then use the tools provided by Kitti-Tracking dataset to project the 3D points onto color images and define that the points in 2D labeled boxes are ground truth.

In Tab. II, method of [8] gets slight advantages over our algorithm in precision of pedestrian segmentation, but the proposed algorithm has advantages over other items in the table. Compared with the methods of [8], our algorithm is relatively insensitive to the threshold of parameters. A disadvantage of the proposed algorithm is that it may divide a large planes (like walls) beside the road into several parts.

TABLE II. Comparison Of Results On Kitti-Tracking

object	method	MaxF	PRE	REC	Runtime
pedestrian	Stachniss [8]	85.15%	**90.08%**	80.74%	150ms
	ours	**86.06%**	88.34%	**83.90%**	200ms
car	Stachniss [8]	84.50%	79.75%	89.18%	150ms
	ours	**86.07%**	**80.14%**	**92.96%**	200ms
cyclist	Stachniss [8]	83.17%	90.72%	76.79%	150ms
	ours	**85.52%**	**93.70%**	**78.65%**	200ms

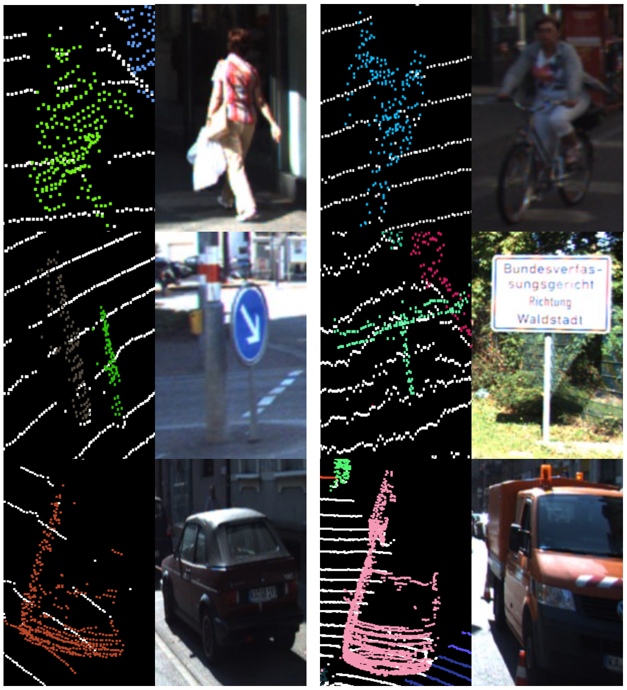


Fig.6 Segmentation result of different objects..

There is a big gap between the runtime in the Tab. II and [8]. The main reason is that the experimental platform and the experimental data of the algorithm are different. The runtime in Tab.II only indicates that our algorithm has the same computational complexity as [8]. The main reason for the long operation time of the proposed algorithm in the table is that we use 8-neighborhood search, while the algorithm of [8] uses 4-neighborhood search.

In Fig.6, we can see that the objects like pedestrians, cyclists and traffic signs are correctly segmented independently. And the proposed connected neighborhood segmentation algorithm works well when dealing with different types of cars.

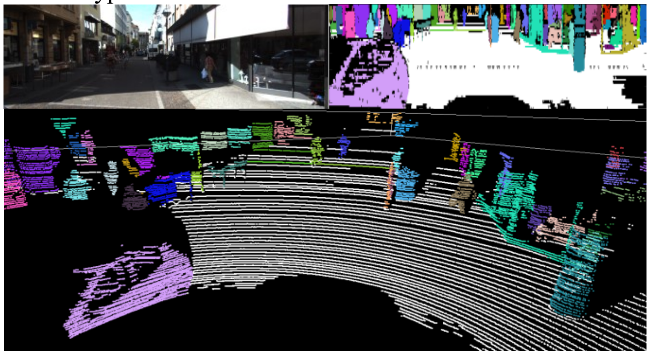


Fig.7 Segmentation result of a complex scene with many kinds of objects.

Fig.7 shows ground and non-road segmentation results together. We display the experimental results in three different forms: 2D color image (top left), 2D lidar-image (top right) and 3D point clouds (bottom).

IV. Conclusion

In this paper, we proposed a two-step unsupervised learning-free 3D point cloud segmentation algorithm. We project the 3D point cloud onto the 2D lidar-image to simplify the calculation. Since the ground detection is the challenge of the whole segmentation, we use improved lidar-histogram algorithm to extract ground points. And then we use the neighborhoods relation given by the lidar-image to connect and cluster points lying on the same object. Experiments valid that the proposed algorithm works well on real urban environment data.

In the future work, we will use CNN model to recognize each object after unsupervised segmentation to achieve panoptic segmentation in 3D point cloud. We believe that the pipeline of step-by-step processing combining with deep learning technology is more practical than the end-to-end pipeline at this stage when to understand complex urban environment for a UGV.

References

[1] A. Dewan, T. Caselitz, G.D. Tipaldi, and W. Burgard. Motion-baseddetection and tracking in 3d lidar scans. In Proc. of the IEEE Int. Conf. on Robotics & Automation, 2016.

[2] P. Y. Shinzato, D. F. Wolf, and C. Stiller. Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion. In Intelligent Vehicles Symposium Proceedings, 2014 IEEE pp. 687-692.

[3] X. Hu, F. S. A. Rodriguez, and A. Gepperth. A multi-modal system for road detection and segmentation. In Intelligent Vehicles Symposium Proceedings, 2014 IEEE, 2014, pp. 1365-1370.

[4] L. Chen, J. Yang, and H. Kong. Lidar-histogram for fast road and obstacle detection. In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2017.

[5] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In Proc. of the IEEE Int. Conf. on Robotics & Automation, 2011.

[6] B. Douillard, J. Underwood, V. Vlaskine, A. Quadros, and S. Singh. A pipeline for the segmentation and classification of 3d point clouds. In Proc. of the Int. Symposium on Experimental Robotics, 2014.

[7] D. Korchev, S. Cheng, Y. Owechko, and K. Kim. On real-time lidar data segmentation and classification. In Proc. of the Intl. Conf. on Image Processing, Computer Vision, and Pattern Recog., 2013.

[8] I. Bogoslavskyi, C. Stachniss. Fast range image-based segmentation of sparse 3D laser scans for online operation. IEEE International Conference on Intelligent Robots and Systems, 2016 pp.163-169.

[9] F. Moosmann, O. Pink, and Ch. Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. In Proc. of the Intelligent Vehicles Symposium, 2009 pp. 215-220.

[10] Kirkland E J. Bilinear Interpolation[M]// Advanced Computing in Electron Microscopy. Springer US, 2010:261-263.

[11] J. Fritsch, T. Kuhnl, and A. Geiger, A new performance measure and evaluation benchmark for road detection algorithms, International IEEE Conference on Intelligent Transportation Systems, 2013, pp. 1693-1700.

[12] Chen Xiaozhi, Ma Huimin, Wan, Ji, Li Bo, Xia, Tian. Multi-View 3D Object Detection Network for Autonomous Driving. IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp.6526-6534

[13] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3D Proposal Generation and Object Detection from View Aggregation. IEEE International Conference on Intelligent Robots and Systems, 2018.

[14] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, Leonidas J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[15] Bo Li. 3D Fully Convolutional Network for Vehicle Detection in Point Cloud. IEEE International Conference on Intelligent Robots and Systems, 2017, pp.1513-1518.