The Ground Segmentation of 3D LIDAR Point Cloud with the Optimized Region Merging

Kiin Na, Jaemin Byun, Myongchan Roh, and Bumsu Seo
Intelligent Cognitive Technology Research Department
ETRI
Daejeon, Republic of Korea
Email: kina4147, jaemin.byun, mcroh, and bsseo@etri.re.kr

***Abstract*—This paper represents a additional approach to enhance the result of ground segmentation method with the gathered point cloud from 3D LIDAR. In this segmentation process, the over-segmentation is usually occurred due to the characteristics of 3D LIDAR such as noise, occlusion, and straightness in complex urban environment. In addition, it has a fatal influence on the entire performance of the perception. In this paper, the region merging algorithm for 3D LIDAR point cloud is proposed to integrate overly partitioned ground regions, which are obtained through the region growing algorithm. First, the initial ground is determined by the current vehicle pose, and then the partitioned regions are ordered according to the distance to the vehicle. In this order, both the ground, where the vehicle is able to reach and the respective region are resampled to pairs of the closest edge pixels. If the resampled edge pixels are satisfied with the region merging criterion, the ground region can merge with the compared region and can expand. This process is iterated until all of the partitioned regions are inspected. The proposed region merging algorithm is demonstrated with the labeled simulation data and the real 3D LIDAR data, as compared to the segmentation method without the propsed region merging.**

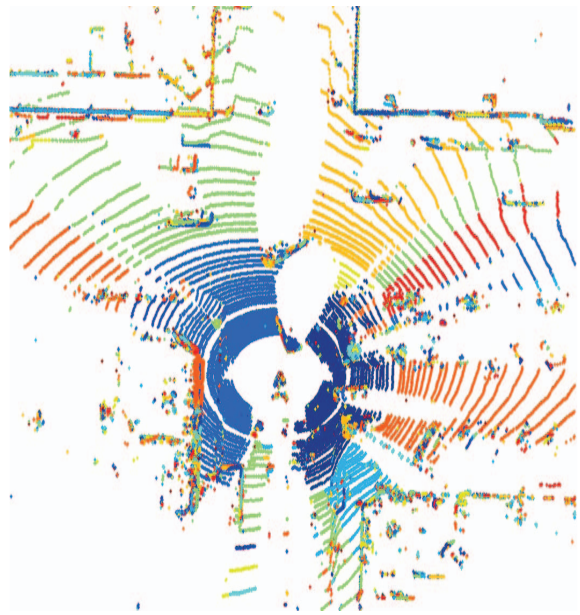

Fig. 1. over-segmenation with 3D LiDAR point cloud; different colors means seperated regions by region growing segmentation algorithm.

I. Introduction

The perception of the robotic system such as the autonomous vehicles is the fundamental task which is related to the entire system. Specifically, in the autonomous vehicle, the surrounding perceptions for obstacles, road marks, traffic lights, traffic signs, and others are critical to its performance and reliability [1]. Nevertheless, the perception in the dynamic and complex environment is one of the main unsolved issues in the robotics field even though it has been researched for a long period.

Usually, the vision sensor has been employed to perceive the surrounding, but it is easily affected by environmental conditions related to light [2]. Therefore, it is not conditionally suitable to apply to the outdoor perception. To develop the robust perception system for various environments, 2D LIDAR has integrated with the vision sensor [3], [4]. However, 2D LIDAR cannot cover all ranges of the vision sensor. In addition, without the vision sensor, one layer of 2D LIDAR is insufficient for the reliable autonomous driving [5]–[7].

The 3D LIDAR for the outdoor perception has been introduced with the open libraries for processing point cloud data [8]. As the point cloud obtained from 3D LIDAR is plentyful and also robust to various environmental conditions, it has been generally applied for the recent developments of the autonomous vehicle [9]–[11]. However, the amount of point cloud data from 3D LIDAR is so abundant that it is not easy to process in realtime. Additionally, occlusion caused by the straightness of laser is also critical enough to interrupt the perception performance of 3D LIDAR .

Generally, the perception algorithm progressively consists of several steps such as segmentation, tracking, classification, and others. In these steps, the segmentation, which is the process partitioning data into multiple parts with specific critierion, is a basic pre-processing in the perception [12]. The major issues of segmentation are over-segmentation and under-segmentation, dividing into excessively many and few parts as shown in Fig. 1 [13]. These problems are critical to entire performance of perception because the results of segmentation could have a strong influence to the overall steps such as tracking and classification .

The objective of this paper is to decrease the number of the overly segmented parts in the ground segmentation with the region merging algorithm. Firstly, the projected 3D LIDAR point cloud onto range image is partitioned with region growing method according to reachability feature criterion and edge pixels of them are also extracted. Secondly, the overly

978-1-4799-2491-2/13/$31.00 ©2013 IEEE

DOI 10.1109/ICCVE.2013.115

segmented regions are ordered according to their distance from the vehicle. They are sequentially inspected with the region merging algorithm. To evaluate the degree of mergence of regions with the ground, close edge pixels between the inspecting region and the ground are respectively extracted to decrease computational cost and to compare only correlated pixels between them. Finally, the satisfied regions with the region merging criterion are merged into the ground, repeatedly. With this algorithm, all the overly segmented grounds are merged into the single ground region and it can provide the well ground removed data to improve the result of the object segmentation.

The remainder of this paper is organized as follows. Sect. II introduces the recent works related to 3D LIDAR based segmentation. Sect. III proposes the enhanced segmentation algorithm with the region growing and the region merging. Sect. IV provides the results of simulation and real experiments to demonstrate the effectiveness of the proposed algorithm. Finally, this paper is concluded with future works in Sect. V.

II. Related Works

The point cloud segmentation has been widely researched for perception. Various segmentation methods have been proposed according to sensors, application, etc. With MRF(Markov Random Field) model, the graph cut algorithm such as normalized cut, min cuts, etc. partitions point cloud data into minimizing the energy function [14]. LBP(Loopy Belief Propagation), a iterative message passing algorithm for minimizing the free energy, is employed for partitioning the reachable region, the drivable region, and the obstacle regions [15]. In addition, minimum spanning tree approach and recursive cutting algorithm for the segmentation of point cloud is proposed in [16].

Moosmann *et al*. propose the region growing based segmentation with a local convexity criterion which consists of the weighted normal and the connectiveness in the mesh based range image. The grounds and the objects are partitioned altogether at the same segmentation process [9], [13].

Douillard *et al*. propose a set of segmentation algorithms for density of data(dense or sparse) and model of data(voxel grid and mesh based). The performance of the segmentation algorithms, GP-INSAC(Gaussian Process Incremental Sample Consensus), mesh based local convexity and cluster-all segmenation are evaluated and compared using hand-labeled data [10].

Himmelsbach *et al*. partition point cloud into the ground to be estimated by simple comparisons to local line fit model and incremental algorithm for realtime performance. Moreover, the remained points of the occupancy grid and the voxel grid are clustered for object detection [11].

The point cloud is overly partitioned with common segmentation approaches in complex environment. It would seriously affect to the result of the overall perception process. Even though region merging method have been employed for solving over-segmentation problem in image processing, it is usually applied when the overly partitioned neighbor segments are satisfied with the image based region merging criterion [17], [18]. This paper contributes to propose the region merging method to solve over-segmentation issue of 3D LIDAR point cloud. The proposed region merging method includes the region partitioning, region merging, and optimization of their parameters to integrate the overly separated ground regions rapidly and efficiently.

III. Ground Segmentation

The ground, in this paper, means the area where the vehicle is possible to reach from current position. Therefore, region growing is employed with the reachability feature criterion composed of gradient, normal difference and height difference. Moreover, the suggested region merging implements to unite the qualified regions in close order to the current vehicle position with the integration feature criterion.

A. Scan Data Projection onto Range Image

Point cloud gathered from 3D LIDAR can be transformed to the various types of model such as ground plane 2D grid, voxel 3D grid, range image, and others. For 2D grid type, point cloud is projected onto 2D grid map and the cells overlapped with points are filled with the representative value such as the highest or average value. As above, the voxel 3D grid type is made up with cubic cells, called voxels which is occupied with 3D points. Since their performances can be scaled by the number of cell which is adjusted by perception area and cell size, it is not appropriate in realtime process for the detailed perception because of a huge number of cell.

In this study, range image with the spherical projection model is applied to easily make point-to-point connection between neighborhood points. To decrease aan influence of noise, the projected image is also smoothed by median and bilateral filter. The range Image can support more rapid process in less data loss for 3D LIDAR point cloud. Moreover, the image process algorithm can be applied to handle point cloud without difficulty.

B. Region Growing based Ground Segmentation

The main purpose of region growing segmentation is to partition input point cloud data into several meaningful regions. This approach examines the neighbors of random seed point and determines if neighbors are added to the region. If the neighbors belong to the region, their neighbors are also inspected repeatedly until no more data is added. After finishing the region growing for one region, another region is expanded again with another seed point. This process operates until no more data is remained. The region growing algorithm is easy to implement and applicable to employ the multiple criterion at the same time. Moreover, it is also suitable to inspect reachable regions from current position.

In this paper, seed points are randomly selected out of unlabeled points and regions are expanded from the seed points toward four neighbors as shown in Fig. 2 while region membership criterion is satisfied. When expansion of region from single seed point is finished, the points in the region are labeled as same number. This process is repeatedly executed until all the points are labeled.

The type of features and their combination in region membership criterion are critical to enhance segmentation

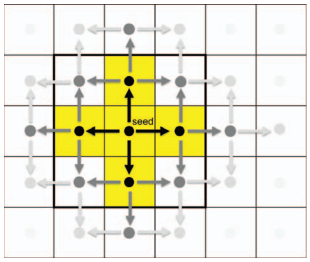

Fig. 2. The region growing toward four neighborhood; This process repeatedly keeps inspecting unlabeled pixels out of four neighbors and the satisfied cells with region growing criterion are labeled with same segment. Black bold boundary means eight neighbors to estimate normal vectors for middle point of them

results. For the ground segmentation, normal difference, height difference and gradient are used to consider where is reachable from seed points by the vehicle [9], [10]. The normal vector, $\vec{n_i}$ of local surface on a point is estimated using eight neighbors as shown in Fig. 2. The derived normal vectors are alligned toward LIDAR on the vehicle. The normal difference, $dn_{i,j}$ is obtained by inner product of two normal vectors; subscript i means current point index, j means one of neighborhood indices and i, j means correlation between two neighbor points. The gradient, $g_{i,j}$, means a inclination between two points. It is then obtained as absolute height difference is divided by distance between two consecutive points.

Due to the characteristic of 3D LIDAR, the closer points to the sensor are, the larger gradient or normal vector noise can be existed because of small amount of sensor noise [10]. Therefore, derived feature values are smoothed by average filter and the region growing criterion is alternated with the height difference, $dh_{i,j}$ within the distance threshold d_{thres}. Finally, the region growing criterion is composed of the normalized features by sigmoidal function and their combination as follows.

$$f_{i,j} = \begin{cases} dh_{i,j} & \text{if } \parallel \vec{d}_{i,j} \parallel \leq d_{thres} \\ \min(g_{i,j}, dn_{i,j}) & \text{otherwise.} \end{cases} \tag{1}$$

C. Ground Region Merging

In segmentation, both over-segmentation and under-segmentation are usually occurred regardless of partitioning methods. The ground segmentation through region growing algorithm has also these problems because of sparse data, sensor noise, occlusion, non-optimal region growing criterion and others. Even though it is difficult to partition point cloud without erroneous segments, under-segmentation is more perilous on ground segmentation for autonomous vehicle because obstacles is also possible to be removed as the ground segment. Therefore, it is better to merge overly separated segments with region merging as shown in Fig. 1.

The over-segmentation by occlusion tends to be severely caused in the complicate urban environment because there are many obstacle to block straight laser. Accordingly, although points actually belong to same object, they could not directly connect to each others and could be separated into different regions. The ground region merging algorithm is proposed to effectively extract ground points as the solution of the over-segmentation problems. In complex urban environment, to represent the ground with single model is not appropriate. Therefore, in this paper, the ground is defined as the area where the vehicle is reachable, not the flat surface. Firstly, the ground is initially generated by the virtual surface where the vehicle locates and the separated regions are ordered by distance from the initial ground to determine inspection sequence.

Secondly, boundary pixels of both the inspecting region and the ground are extracted as the representative points. Actually, they are descriptive enough to estimate similarity to the ground because they already include some estimated features such as normal difference, gradient and height difference. Moreover, all the points in same region are connected with them by region growing criterion. To reduce the number of samples and to properly compare points in different regions, the nearest cells in different boundaries are paired using kd-tree, which one cell of pairs is not even overlapped to other pairs as shown in Fig. 3.

Lastly, homogeneity and continuity of the regions with the ground are estimated to determine integration with inspecting regions. To compare one of nearest pairs, the points within specified pixels from the nearest point in same segment are chosen. With these points, the average normal vector and the average position are respectively calculated and features such as normal vector, gradient, and height difference are derived. In this step, once even one of nearest pairs is satisfied with region merging criterion in Eq. 2, the current ground absorbs the inspecting regions and the edges of the expanded ground are also updated. It is no matter that two regions determine integration with only single approachable pair because other pixels in same region are already connected with edges. Furthermore, two edges in pair are also estimated that they are on same surface. Even though integration of two regions can be estimated by the ratio of qualified pair, it is not reasonable that connected area of regions are on same surface. This process repeatedly continues untill all the regions are examined if they are merged with the ground.

$$f_{i,j} = \min(g_{i,j}, dn_{i,j}) \tag{2}$$

IV. Experiment

To demonstrate the proposed segmentation method, the two ways of experiments are carried out with the simulation data of PreScan and the real data of velodyne HDL-64E S2. To clearly check the segmentation result, data within 120 degree of the front is applied to the segmentation process. The results with and without the proposed region merging algorithm are compared to prove an enhancement of the result of ground segmentation method. In the ground segmentation without region merging, the ground is selected the closest and the biggest segment from the vehicle. In this experiment, the type of segmentation methods is not a critical issue because they have similar results and over-segmentation is physically happened by occlusion of points. Therefore, the results with and without region merging have a difference according to the degree of occlusion and noise of sensor data. The proposed region merging algorithm is developed by MATLAB.

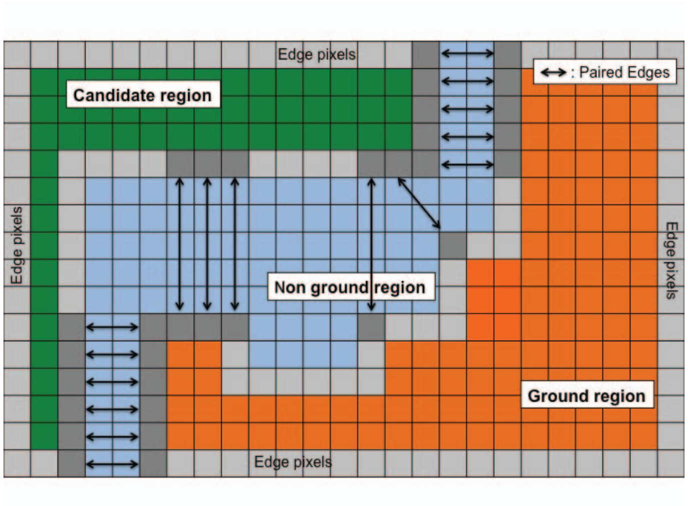

Fig. 3. The example of region merging algorithm; the ground and the candidate with different colors are seperated with non-ground region. Bright gray cells mean edge pixels of each regions, dark gray cells mean selected pixels as paired edges based on distance, and bi-directional arrows mean that connected two cells are paired.

Input: $P^{edge}_{ground}, Candidates = \{P^{edge}_{region_1}, P^{edge}_{region_2}, ..., P^{edge}_{region_n}\}$
Output: P_{ground}
for $P^{edge}_{region_n} \in Candidates$ **do**
  $NearestPairs \leftarrow$ getUniqueNearestPair($P^{edge}_{ground}, P^{edge}_{region_n}$);
  for $[i, j] \in NearestPairs$ **do**
    if *isConnect*($P^{edge}_{ground}(i), P^{edge}_{region_n}(j)$) **then**
      $P^{edge}_{ground} \leftarrow P^{edge}_{ground} \cup P^{edge}_{region_n}$;
      $P_{ground} \leftarrow P_{ground} \cup P_{region_n}$;
      break;
    end
  end
end

Algorithm 1: The process of the region merging algorithm with the ground; i is a index of the ground and j is a index of the region.

A. Parameter Setting

In this paper, several parameters are used as threshold values in region growing and region merging such as dh^{grow}_{thres}, g^{grow}_{thres}, dn^{grow}_{thres}, g^{merge}_{thres}, and dn^{merge}_{thres}. These parameters are set up by particle swarm optimization(PSO), which is one of the the optimization process, with different types of ground truth data [20]. PSO can optimize non-linear problem by iteratively enhancing a candidate solution according to value of the cost function which tends to be changed by given parameters. This optimization process has a population of particles as a candidatre solution and these particles move within the defined search space over their position and velocity. The randomly initialized particles tend to converge toward one of the particles, which has the best parameter values. In this paper, PSO is applied to optimize region growing and merging results altogether because these are not fomulated by the applied threshold parameters as shown in Fig. 4.

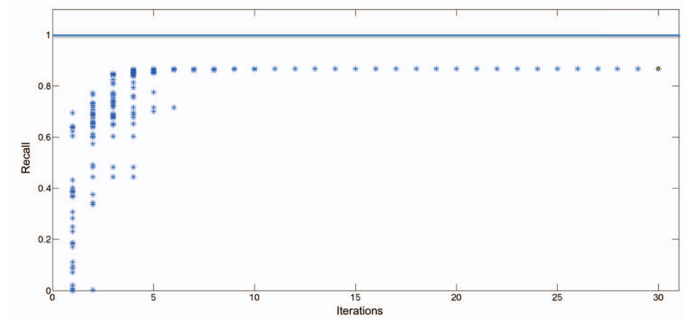

Fig. 4. The particle swam optimation process; the thirty particles are guided to the optimal solution in thirty interations.

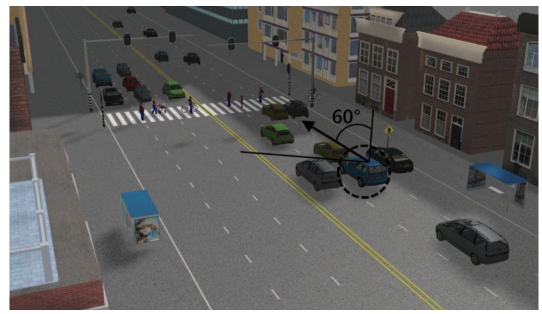

Fig. 5. PreScan simulation environment;the vehicle in a dashed circle is equipped with the modeled 3D LIDAR; lines represent both direction and FOV of 3D LIDAR.

B. Simulation

To test the proposed algorithm in simulation environment, PreScan has been utilized. PreScan is a software tool that can be used for developing advanced driver assistance systems(ADAS) based on sensor technologies such as RADAR, LIDAR, camera and GPS, etc. The simulation environment is set as shown in Fig. 5. The 3D LIDAR with 64 layers and 120 degree as the horizontal FOV is modeled as the velodyne sensor. The labeled simulation data into the ground and the non-ground is obtained as shown in Fig. 6(a). The threshold parameters are fixed to $dh^{grow}_{thres} = 0.04m$, $g^{grow}_{thres} = 0.13$, $dn^{grow}_{thres} = 0.3$, $g^{merge}_{thres} = 0.16$, $dn^{merge}_{thres} = 0.3$ by optimization process. Firstly, the point cloud is partitioned into several regions by the region growing method. The result without and with the suggested region merging algorithm is as shown in Fig. 6(b) and in Fig. 6(c).

The recall value is calcluated to check the performance of segmtentation [21]. The results of the ground segmentations are reported in Table I. Since the point cloud from the simulation environment are physically separated into two large regions, two results have a big difference. According to simulation results, it is noticeable that the ground segmentation without region merging is strongly influenced by the input data and the environment.

C. Real Experiment

To gather point cloud data of outdoor environment, the vehicle equipped with velodyne sensor is used as shown in Fig.

TABLE I. THE ACCURACY OF TWO TYPES OF GROUND SEGMENTATION METHODS USING SIMULATION DATA

Method	Recall
ground segmentation without region merging	77.40
ground segmentation with region merging	99.39

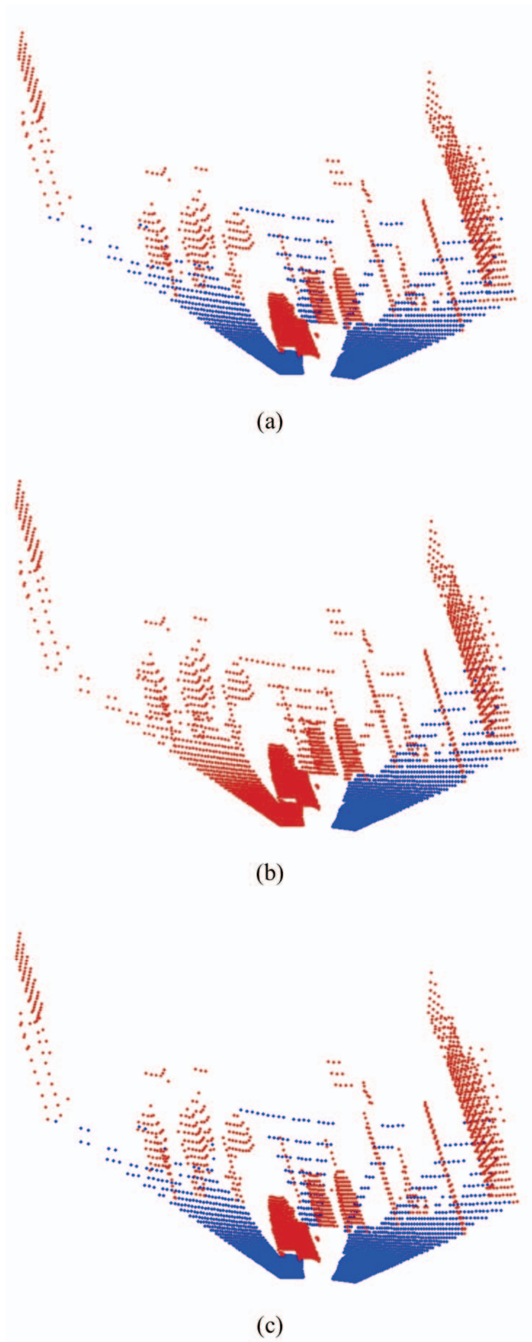

Fig. 6. (a) The labeled data from PreScan simulator; blue color parts belong to the ground; another color, red, indicates the non-ground segments. (b) the biggest and the closest segment as the ground, without region merging. (c) the ground segmentation with the proposed region merging algorithm; the regions in the middle and the left side are merged with the ground.

Fig. 7. The vehicle for gathering 3D LiDAR point cloud with velodyne HDL-64E S2.

7. The applied velodyne sensor consists of 64 laser diodes and rotates at a rate of 10Hz. At each turn, about 120,000 points of the upper 32 lasers and the lowers 32 lasers are gathered separately with the different number of shots. To project points onto range image, point cloud can be projected on the spherical coordinate system. Moreover, disconnected or empty cells are linearly interpolated to utilize smooth and dense range image.

To prove the performance of the proposed methods, the ground truth is made up as the ground and the non-ground pixels are respectively set with different labels. Some parts of this dataset are used to optimize the parameter by PSO, and other parts are applied to evaluate the performance of the proposed region merging. The threshold parameters are fixed as $dh^{grow}_{thres} = 0.05m$, $g^{grow}_{thres} = 0.23$, $dn^{grow}_{thres} = 4.32$, $g^{merge}_{thres} = 0.16$, $dn^{merge}_{thres} = 3.16$ by opimization process.

As shown in Fig. 8(a), the result of segmentation with region growing has a lot of partitions. In these partitions, the ground as the biggest and the closest segment in Fig. 8(b) were selected without the suggested region merging algorithm. The physically separated right side of grounds is not merged to the method without region merging. However, with the proposed region merging algorithm, these right side of the regions are also included as the ground. The enhancement of the proposed region merging method can be visually checked as shown in Fig. 8(c). This result can be also checked with Table II

TABLE II. THE ACCURACY OF TWO TYPES OF GROUND SEGMENTATION METHODS USING 3D LIDAR DATA

Method	Recall
ground segmentation without region merging	63.14
ground segmentation with region merging	92.53

V. CONCLUSION

This paper has proposed the region merging algorithm for improving the result of 3D LIDAR point cloud ground segmentation. The ground has been partitioned with the region growing algorithm, but regions were overly segmented due to the characteristic of 3D LIDAR and a complex urban enviroment. In addition, it has bad effects on the object segmentation. Therefore, the proposed region merging for 3D LIDAR point cloud was employed to join the overly partitioned ground regions. The proposed algorithm was tested with the labeled simulation data and the velodyne sensor data, as compared to the ground segmentation without region merging. In the future, this approach will be extended to a overly partitioned object segments. In addition, both the ground and the object segmentation results will be also improved with tracking results. Moreover, the proposed methods will be implemented on the realtime system such as autonomous vehicle.

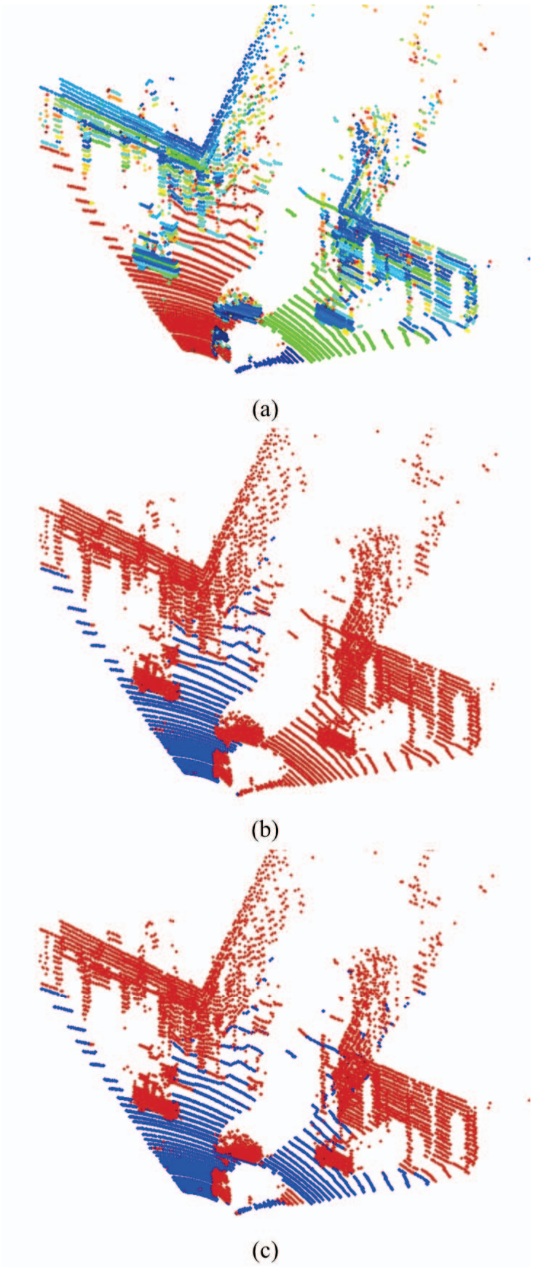

Fig. 8. (a) The overly partitioned data from 3D LIDAR; each region is colored with different colors. (b) the biggiest and the closest segment as the ground (c) the ground segmentation with the proposed region merging algorithm; the right side of regions are merged with the ground, unlike the method without region merging.

Acknowledgment

This work was supported by the IT R&D program of MSIP/KEIT. [KI0041417, Development of Decision Making/Control Technology of Vehicle/Driver Cooperative Autonomous Driving System (Co- Pilot) Based on ICT].

References

[1] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun (2011) Towards Fully Autonomous Driving: Systems and Algorithm. In: The IEEE Intelligent Vehicles Symposim. 163–168.

[2] A. Ess, K. Schindler, B. Leibe, and L. Van Gool (2010) Object detection and tracking for autonomous navigation in dynamic environments. The International Journal of Robotics Research. 29(14): 1707–1725.

[3] L. Spinello, R. Triebel, and R. Siegwart (2008) Multiclass Multimodal Detection and Tracking in Urban Environments. The International Journal Robotics Research (IJRR). 3081

[4] Q. Baig, O. Aycard, T. D. Vu, and T. Fraichard (2011) Fusion Between Laser and Stereo Vision Data For Moving Objects Tracking In Intersection Like Senario. In: IEEE Intelligent Vehicles Symposium. 362–367.

[5] K. Schueler, T. Weiherer, E. Bouzouraa, and U. Hofmann (2012) 360 Degree Multi Sensor Fusion for Static and Dynamic Obstacles. In: IEEE Intelligent Vehicles Symposium. 692–697.

[6] F. Nashashibi and A. Bargeton (2008) Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation. In: The IEEE Intelligent Vehicles Symposium. 847-852.

[7] R. MacLachlan (2005) Tracking Moving Objects From a Moving Vehicle Using a Laser Scanner. Technical Report CMU-RI-TR-05-07, Robotics Institute, Carnegie Mellon University.

[8] R. Bogdan Rusu and S. Cousins (2011) 3D is here: Point Cloud Library (PCL). In: IEEE international conference on Robotics and automation (ICRA).

[9] F. Moosmann and T. Fraichard (2010) Motion estimation from range images in dynamic outdoor scenes. In: The IEEE International Conference on Robotics and Automation (ICRA). 142-147.

[10] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A, Frenkel (2011) On the Segmentation of 3D LIDAR Point Clouds. In: The IEEE International Conference on Robotics and Automation (ICRA). 2798-2805.

[11] M. Himmelsbach and Felix v. Hundelshausen and H.-J. Wuensche (2011) Fast Segmentation of 3D Point Clouds for Ground Vehicles. In: The IEEE Intelligent Vehicles Symposium. 2798-2805.

[12] K. Klasing, D.Wollherr, and M. Buss (2008) A clustering method for efficient segmentation of 3D laser data. In: The IEEE International Conference on Robotics and Automation (ICRA). 4043-4048.

[13] F. Moosmann, O. Pink, and C. Stiller (2009) Segmentation of 3D Lidar Data in non-flat Urban Environments using a Local Convexity Criterion. In: The IEEE Intelligent Vehicles Symposium. 215-220.

[14] Aleksey Golovinskiy and Thomas Funkhouser (2009) Min-Cut Based Segmentation of Point Clouds. In: The IEEE International Conference on Computer Vision Workshops. 39–46.

[15] C. Guo, W. Sato, L. Han, S. Mita, and D. McAllester (2011) Graph-based 2D Road Representation of 3D Point Clouds for Intelligent Vehicles. In: The IEEE Intelligent Vehicles Symposium. 715-721.

[16] C. S. R. Aguiar, S. Druon, and A. Crosnier (2007) 3D datasets segmentation based on local attribute variation. In: The IEEE/RSJ International Conference on Intelligent Robots and Systems. 3205-3210.

[17] B. Peng, L. Zhang, and D. Zhang (2011) Automatic Image Segmentation by Dynamic Region Merging. In: The IEEE Transactions on Image Processing. 3592-3605

[18] S. R. Vantaram, and E. Saber (2006) Survey of Contemporary Trends in Color Image Segmentation. Journal of Electronics Imaging.

[19] K. Klasing, D. Althoff, D.Wollherr, and M. Buss (2009) Comparison of surface normal estimation methods for range sensing applications. In: The IEEE International Conference on Robotics and Automation (ICRA). 3206-3211.

[20] D. Bratton and J. Kennedy (2007) Dening a standard for particle swarm optimization. In: Swarm Intelligence Symposium. 120-127.

[21] J. Fritsch and T. Kuehnl and A. Geiger (2013) A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In: International Conference on Intelligent Transportation Systems (ITSC).