AN OBJECT SEGMENTATION APPROACH FOR MOBILE LIDAR POINT CLOUDS

(MOBİL LİDAR NOKTA BULUTLARI İÇİN BİR NESNE AYIRT ETME YAKLAŞIMI)

by

Musa Şervan ŞAHİN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Jan 2021

ACKNOWLEDGEMENTS

My thanks to my Advisor and Galatasaray University Laboratory Technologies of Information and Communication (LabTIC) for its great work environment and computational facilities.

Musa Şervan ŞAHİN
January 2021

TABLE OF CONTENTS

LIST OF SYMBOLS

PCD : Point Cloud Data
LiDAR : Light detection and ranging
DBSCAN : Density-based spatial clustering of applications with noise
ANN : Artificial neural network
CNN : Convolutional neural network
DGCNN : Dynamic graph convolutional neural network
KNN : K-nearest neighbors
GPU : Graphics processing unit
CPU : Central processing unit
FCM : Fuzzy c-means algorithm
PCM : Possibilistic c-means algorithm
RANSAC : Random sample consensus
CRF : Conditional Random Fields
FPFH : Fast Point Feature Histograms
ER-CCL : Elevation-Reference Connected Component Labeling

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

LiDAR sensors are used to percept a scene by its 3D point cloud data. But LiDAR sensor measurement data are not homogeneous, they have uneven density and subject to occlusion problems. And the technical challenge lies along object segmentation because well-known clustering methods can not perform well with stand-alone 3D point cloud data. In this study, an approach is proposed by defining two classes of objects that are ground and nonground objects. The points are labeled ground and nonground points for DBSCAN based method. Then, sensitivity is adjusted for each class to improve the segmentation performance of ground and non-ground objects. For the CCL based method, three bird's eye view matrices are derived from point cloud data. The first matrix is used to separate ground and nonground areas and to merge other two matrix results. The ER-CCL algorithm applied with a smaller sided cell for segmentation of ground objects and a bigger sided cell for segmentation of nonground objects. The presented approach is benchmarked with a DBSCAN based method and a CCL based method. The presented method outperforms the DBSCAN method by 4% and 13% and outperforms the CCL based method by 13% and %20 in v-measure and completeness clustering metrics, respectively. Overall, an approach is proposed to segment 3D terrains in an urban area with different segmentation methods.

Keywords: 3D object segmentation, clustering, LiDAR, DBSCAN

ÖZET

LiDAR sensörleri bir sahneyi 3 boyutlu noktalarıyla algılamak için kullanılan araçlardan birisidir. Fakat bu sensörlerden elde edilen nokta bulutları homojen değildir: Noktalar değişken yoğunluklara sahiptir ve gölgelenme problemleri mevcuttur. Bu nedenden dolayı nesne ayırt etme problemi ortaya çıkar çünkü bilinen kümeleme yöntemleri tek başına 3 boyutlu noktalarda kötü sonuçlar vermektedir. Bu çalışmada yerde olanlar ve yerde olmayanlar şeklinde iki sınıf tanımlayan bir yaklaşım geliştirildi. DBSCAN temelli yöntem için öncelikle noktalar yerde olan ve olmayan şeklinde etiketledi. Daha sonra ayırt etme hassaslığı her bir sınıf için farklı ayarlandı. CCL tabanlı yöntem içinse nokta bulutlarından üç tane kuş bakışı matris türetildi. Bunlardan ilki yeri ayırt etmek ve diğer iki matrisin sonuçlarını birleştirmek için kullanıldı. Kalan iki matris üzerinde ER-CCL yöntemi kullanıldı. Bunlardan biri uzun kenarlı hücrelerden oluşup yerde olmayan nesneleri ayırt etmek için kullanıldı. Diğeri ise kısa kenarlı hücreler kullanarak yerdeki nesneleri ayırt etmede kullanıldı. Sunulan yaklaşım DBSCAN temelli ve CCL temelli yöntemlerle değerlendirildi. Oluşturulan yöntemlerin v-measure ve completeness kümeleme metriklerinde temel aldıkları yönteme göre sırasıyla DBSCAN için %4 ve %13, CCL için %13 ve %20 oranında daha iyi sonuç verdiği görüldü. Böylelikle kentsel alanlarda 3 boyutlu alanları ayırt edebilmek için farklı ayırt etme yöntemleriyle kullanılabilen bir yaklaşım sunulmuş oldu.

Anahtar kelimeler: 3 boyutlu nesne ayırt etme, kümeleme, LiDAR, DBSCAN

1 INTRODUCTION

3D object segmentation is a computer vision algorithm used to improve perception in many domains such as augmented reality, driving awareness for self-driving vehicles, object detection, object tracking, remote sensing, 3D terrain reconstruction and semantic mapping. LiDAR technology is used to percept 3D environments. Laser beams produce point clouds in a 3D scene. The point cloud data includes list of points where each point includes the (x, y, z) coordinates. The mobile LiDAR point cloud means that the data is gathered from a LiDAR mounted on a travelling ground vehicle. Although Lidar technology challenges other sensors with its capability of measuring at nighttime and being less affected by light severity, the main disadvantage of LiDAR data is inhomogeneity of points. This problem may cause wrong segmentation in clustering algorithms. To refine segmentation, a method is proposed by adjusting sensitivity for ground and nonground class of objects in the measured scene. This approach is applied to a DBSCAN (Ester et al., 1996) based method and a CCL (Tian et al., 2020) based method. In implementation of the DBSCAN based method, firstly ground height threshold is extracted to distinguish between ground and nonground objects and filtering ground points. Then, DBSCAN with 1 meter epsilon is applied to ground filtered point cloud which gives ultimate results for ground objects and completes ground and nonground labelling. Lastly, a post processing is applied to nonground points that merges clusters by 2 meters epsilon. For the CCL based method, some bird's eye view matrices initialized by projection of point clouds. Initially, the height threshold is computed by the same method and it is used to get bird's eye view of ground nonground matrix. This matrix has 0.1-meter sided cells. By down sampling, 2 meters sided cells are produced. On the next step, binary object matrix is created having 0.1-meter sided cells. Two binary object matrices are obtained from this matrix. One of them has 0.5-meter sided cells and other one has 1-meter sided cells. The ER-CCL method is used on these two matrices to get object labels. After that, 0.5-meter sided matrix is used for ground object labelling and other one is used for nonground object labelling. To merge results of two matrices, a 2 meters sided ground matrix is used. Finally, segmentation of point cloud is done by back projection of merged matrix.

The contribution of this study is to evaluate different sensitivity thresholds applied to segment ground points and nonground points extracted from LiDAR measured outdoor scenes.

In the main part of this thesis, Section 2 includes related works for ground segmentation and object segmentation in 3D spaces. Section 3 has detailed explanation of problem. Section 4 contains proposed DBSCAN based method and section 5 has ER-CCL based method. In section 6, experiments, metrics, and results are stated by comparing our methods with their base methods. Finally, conclusion and future work is written in Section 7.

2 LITERATURE REVIEW

Although it is not the main task on this work, ground segmentation is required to use DBSCAN (Ester et al., 1996) and ER-CCL (Tian et al., 2020) for segmentation. Douillard et al. (2014) have developed ExractGroundSurface algorithm on their study which utilized from voxelization and center of mass of voxels. Chu et al. (2019) proposed several algorithms which try to find scan lines of LiDAR. Their method finds start points of scan lines by processing through vertically, after finding start points horizontal processing is executed. Narksri et al. (2018) used two algorithms in a row to segment ground: first algorithm benefits from LiDAR rings and their distances between each other, second one uses multi region RANSAC plane fitting. Velas et al. (2018) have used CNN architecture to solve this problem. They represented PCD features as multi-dimensional 2D matrix where axes are velodyne channels and rotation angle.

For the object segmentation tasks, Nguyen and Le (2013) published a comprehensive survey and they categorized point cloud object segmentation methods as edge-based, region-based, model-based, attributes-based and graph-based methods. However, this survey does not include deep learning methods which are widely used in recent years. If an effective ANN architecture is constructed, predictions can be done faster. Because there is no complex algorithm is running while making predictions: The only operation is getting results from trained ANN. Huang & You (2016) proposed point cloud labeling method by using 3D CNN. Their method minimizes extracted features before segmentation. The method merely extracts 3D voxels from point cloud data to feed ANN architecture. PointNet (Qi et al., 2017b) provides segmentation of unordered point sets and high efficiency on 3D point clouds. It does not require any preparation, directly takes (x, y, z) coordinates of points. Based on PointNet, PointNet++ (Qi et al., 2017a) is developed. This architecture can find local features by using metric space distances.

DGCNN (Wang et al., 2019a) has proposed EdgeConv which is also used on point clouds and can incorporate local neighborhood information.
The edge-based methods try to find borders of 3D objects for segmentation. Bhanu et al. (1986) proposed and compared three approaches for edge detection which are gradient, line fitting and surface normal approaches. Jiang et al. (1996) have shown an algorithm named scan line grouping technique which aids to find curve segments of objects. The object segmentation is done by growing from the curve segments. Sappa & Devy (2001) have made edge detection in two steps. First, a binary edge map is generated benefiting from scan line technique in (Jiang et al., 1996). Then, contour extraction is done by using binary edge map. Edge detection strategies are not useful when point density is decreased.

The region-based methods first find a center then extends this center with neighbors. Besl and Jain (1988) presented a method that consist of two phases: regions are roughly estimated then a connected component algorithm is used to refine regions. The region growing algorithm is equipped with irregular graph pyramid which utilize to get neighbor region information in another study (Koster & Spann, 2000). Pu and Vosselman (2006) extracted building features by using planar surface growing algorithm (Vosselman et al., 2004). The algorithm searching for surface points by looking at whether the point has minimum adjacent points in fitting a plane. Another study which can segment buildings, trees and ground at first stage (Ning et al., 2009) also segments windows, walls and roofs from buildings at the second stage. Dorninger & Nothegger (2007) have used local regression plane fitting parameters in clustering distance matrix while segmentation. This approach reduced time complexity of clustering. Nurunnabi et al. (2012) presented a surface segmentation algorithm based on ROBPCA (Hubert et al., 2005) which is an improved version of well-known PCA.

The attributes-based methods select features of point clouds. After normalization of features, clustering methods are applied for object segmentation. Biosca & Lerma (2008) proposed a cluster merging algorithm which is used with fuzzy algorithms FCM (Peizhuang, 1983) and PCM (Krishnapuram & Keller, 1996). Filin (2002) made a clustering algorithm for surfaces upon airborne laser data. This method improved by slope adaptive neighborhood which assist to get more robust attributes (Filin & Pfeifer, 2006).

The hough transform (Hough, 1962) is adopted to 3D space by Vosselman & Dijkman (2001) to segment planar surfaces from point clouds. Hackel et al. (2017) have used a kind of KNN algorithm to find adjacent points. The Algorithm runs on an area with radius r. The radius value increases until k is reached in KNN. But this customized KNN algorithm has computational difficulty. Wang et al. (2017) created a hash table by projecting 3D point clouds to x-z plane for their method. On the hash table points are clustered according to space between them. Since CPU cannot effectively compute adjacent units, researchers tended to use GPU for this issue. Tian et al. (2020) have proposed elevation reference connected component labelling algorithm with GPU programming which makes clustering while finding connected components. The attributes-based methods give bad results when points are not distributed homogeneously.

The model-based methods utilize from geometric shapes such as plane, cylinder etc. to segment objects. The base method RANSAC is proposed by Fischler & Bolles (1981). and can detect geometric features. Schnabel et al. (2007) used RANSAC on point clouds to detect plane, sphere, cylinder, cone and torus shapes. Gelfand & Guibas (2004) detected slippable shapes which are rotationally and translationally symmetrical. So, different forms of shapes can be extracted. And their algorithm segments objects as combination of slippable shapes. RANSAC and 3D Hough transform methods are compared for segmentation of building roof planes by Tarsha-Kurdi et al. (2007). The RANSAC algorithm gave better result for accuracy it was faster than Hough transform. Li et al. (2011) improved RANSAC results in global aspect. Local results are obtained by RANSAC, after that global relation features such as orientation, placement, and equality are gotten by their algorithm.

The graph-based methods represent point clouds as graph where points are vertexes and edges are neighborhoods between points. In FH algorithm (Felzenszwalb & Huttenlocher, 2004), approximate nearest neighbor algorithm (Arya & Mount, 1993) is used after graph representation. Golovinskiy & Funkhouser (2009) constructed graph by using KNN and proposed a minimum cut based segmentation method. Their algorithm includes a background and foreground discrimination feature for better segmentation. Since an edge between background and foreground has lower weight minimum cut based method tends to cut this kind of edges. Strom et al. (2010) used graph-based approach on colored point clouds. Using additional color information made this study more efficient. Rusu et al. (2009b) used CRF (Lafferty et al., 2001) which is a type of undirected graph for data representation and made surface segmentation utilizing from FPFH (Rusu et al., 2009a). Schoenberg et al. (2010) made an information fusion with laser data and camera data. Markov Random Field (Cross & Jain, 1983) is used to compute corresponding pixel of each point which proposed in another study (Diebel & Thrun, 2005). Since graphs can carry lots of information on them, graph-based methods give efficient results. However, graph operations are time consuming because of complex structure of graphs.

Unlike our study, DBSCAN using studies have used same epsilon value for whole scene. Czerniawski et al. (2018) mapped xyz space to normal vector space as n_1, n_2, n_3. Then, they executed DBSCAN with these six parameters. However, they used constant 0.3 epsilon value. Wang et al. (2019b) proposed a method to estimate optimum epsilon value for LiDAR data. Similarly, even if the estimated value is optimized, it is constant for every part of scene. In (Tian et al., 2020), ER-CCL method also uses boxes with a 5 cm constant size in a projected matrix.

3 ANALYSIS OF MOBIL LiDAR DATA IN SEGMENTATION ASPECT

Some of common methods have a main problem about object segmentation for LiDAR scenes. The problem is using same sensitivity for whole scene gives segmentation errors. When segmentation algorithm is sensitive in clustering, non-ground objects are divided into many clusters. On the other hand, when it is not sensitive, ground objects are merged in same clusters (Figure 3.3). There are several reasons of this error. Ground and non-ground object definitions of this thesis must be stated before explaining the error reasons. Non-ground objects occupy huge area and they have greater height values. Ground objects occupy small area, and they are surrounded by ground. Table 3.1 shows ground and non-ground object instances.

Table 3.1: Ground and non-ground object instances

Ground Objects	Non-ground Objects
Cars, trucks, pedestrians, cyclists, traffic signs	Vegetations and buildings

The first reason of segmentation error is arisen from size difference between ground and non-ground objects. Second, ground objects prevent LiDAR points to reach non-ground objects (Figure 3.1). When they create a shadow, non-ground objects seem like divided. Third, LiDAR point density is decreased by distance. So, away object points are far away from each other and this causes to find more objects than real. In the Figure 3.2, there is a vegetation area on left image. In the middle image, this area is separated into several clusters because of lack of points. However, the vegetation area should be one cluster as seen on right camera image. The area is on the left, behind of signboard.

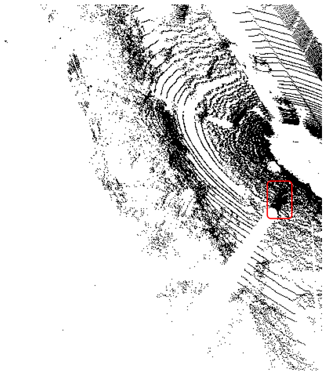

Figure 3.1 Example of object shadow captured by a LiDAR

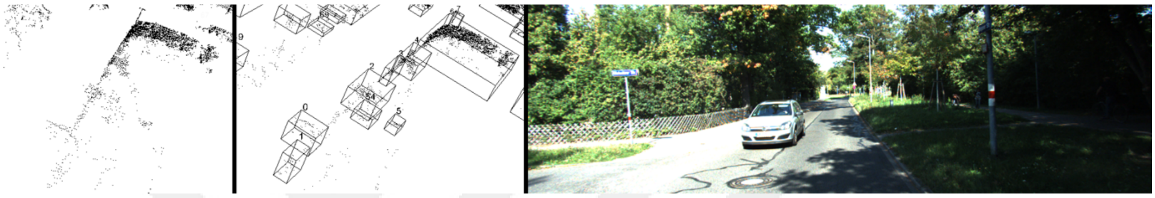

Figure 3.2 Example of decrement of point density by distance

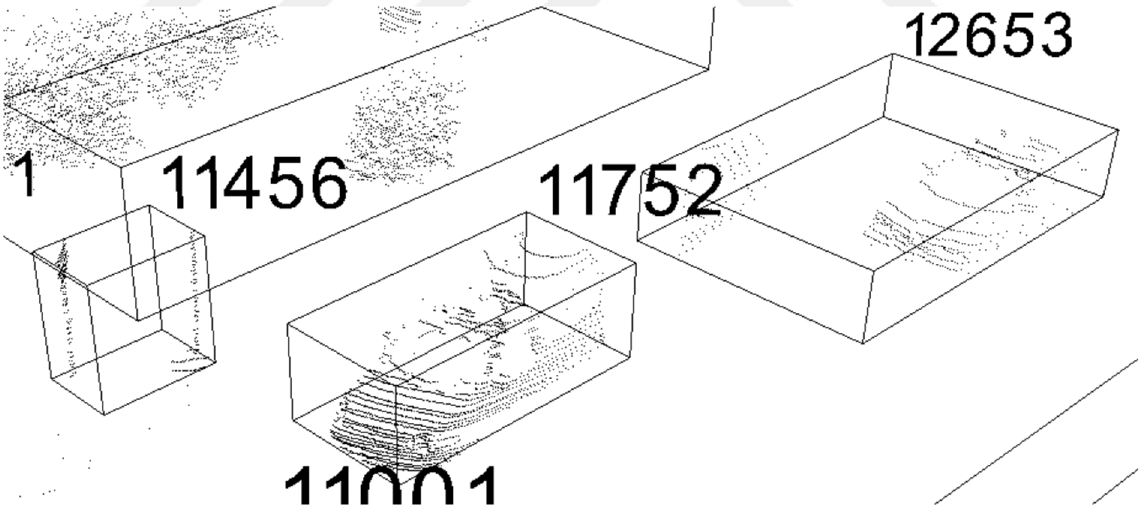

Figure 3.3: Illustration of merging for objects with an ID 11456 and 12653

4 THE DENSITY-BASED SPATIAL CLUSTERING of APPLICATIONS with NOISE BASED METHOD

This method uses DBSCAN algorithm for clustering issue. DBSCAN has two main parameters named minimum points and epsilon. To give different sensitivities for ground and non-ground objects, different epsilon values are used.

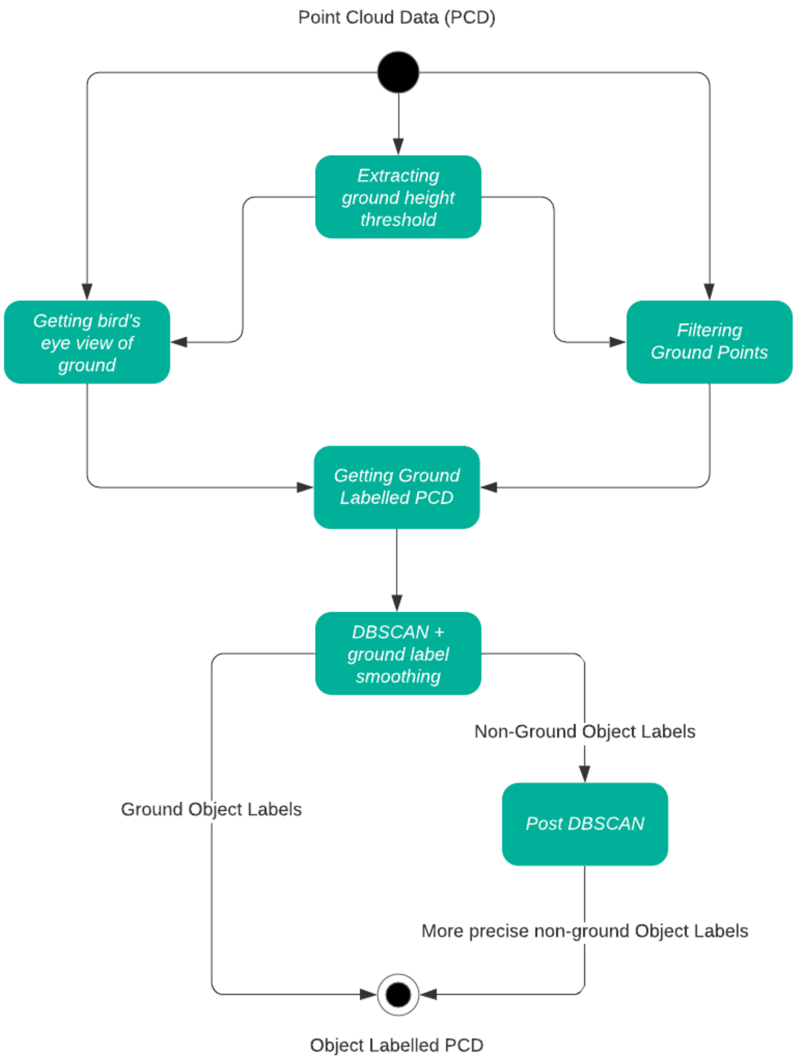

Figure 4.1: DBSCAN based method overview

4.1 Extracting Ground Height Threshold

The height values are sorted in descending order and visualized from the PCD. It is seen that values become more horizontal after a threshold between 0.5 and -1.8. Since ground height values are close to each other but object heights vary. Piecewise linear regression is used to find this threshold. The intersection point of two lines is the extracted height value.

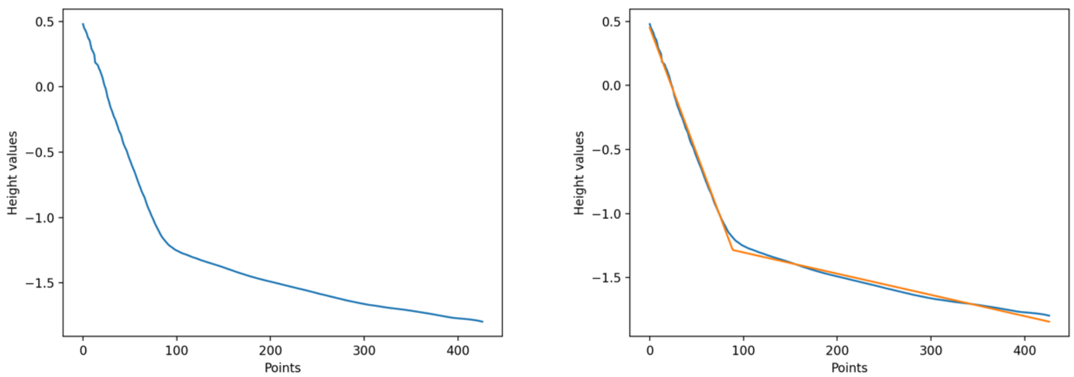

Figure 4.2: The sorted height values between 0.5 and -1.8(on the left). Fitted two linear lines (on the right, orange colored).

4.2 Getting Bird's Eye View of Ground

The scene is projected into 2D matrix. Each cell of matrix has 2 meters side. If cell contains points below height threshold 1 is assigned to cell, otherwise 0 is assigned. 1 represents ground cell, 0 represents non-ground cell. After that, second image is obtained in the Figure 4.3 where ground cells are yellow, non-grounds are blue (the first image is added to imagine scene). To get more clean ground, a recursive algorithm is executed over this matrix. If there are at least 3 ground cells around a non-ground cell, the type of cell is changed to ground cell. Then third image is created in the Figure 4.3.

Algorithm 1. Recursive Ground Cleaning Algorithm

Input: matrix, i: row index, j: column index

Output: Cleaned matrix

```
function assigner(matrix, i ,j):
    if matrix[i][j] = 0 then
        total ← Matrix[i + 1][j] + Matrix[i][j + 1] + Matrix[i - 1][j] + Matrix[i][j - 1]
        if total > 2 then
            matrix[i][j] ← 1
                assigner(matrix, (i + 1), j)
                assigner(matrix, i , (j + 1))
                assigner(matrix, (i - 1), j)
                assigner(matrix, i,  (j – 1))
        end if
    end if
```

Lastly, for the border cells of ground, 2 assigned to get better ground labelling in next phases. As result, the fourth image is created where ground cells are turquoise, border cells are yellow and non-ground cells are blue.

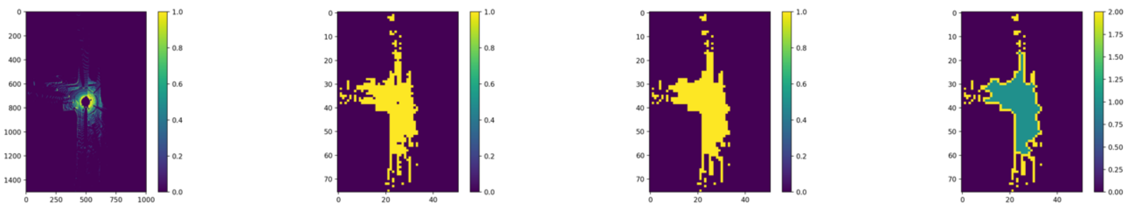

Figure 4.3: The steps while getting bird's eye view

4.3 Filtering Ground Points

Ground points must be removed to do clustering by DBSCAN algorithm. The points below the height threshold are regarded as ground points. After that if a ground point has another ground point in 1cmx1cm cell, the points that have greater height values are removed from ground points. Lastly, ground points are removed from PCD.

4.4 Getting Ground Labelled PCD

For ground filtered points, if point correspond to ground cell in bird's eye view, labelled as ground. Non- ground labels are assigned in the same way. When point is in the border area height value is considered. When height is greater than height threshold point is non-ground, otherwise point is ground point.

4.5 DBSCAN and Ground Label Smoothing

Clustering is done by DBSCAN algorithm with 1 meter epsilon value. If an object includes both ground and non-ground labelled points all of points labelled as non-ground. Ground objects are used for ultimate result. Non-ground points are taken to post processing with object labels.

4.6 Post DBSCAN

In this part, non-ground values are labelled with 2 meters epsilon. For doing this, old object labels from DBSCAN part are used. To speed up this algorithm, kd-tree (Bentley, 1975) is used which is useful for quick neighbor fetching. The post DBSCAN merges the clusters that obtained by 1 meter epsilon and considers outlier points if they should be included to any clusters. Rather than running a new DBSCAN by 2 meters epsilon value, this algorithm is more effective because it uses additional label information from 1-meter DBSCAN.

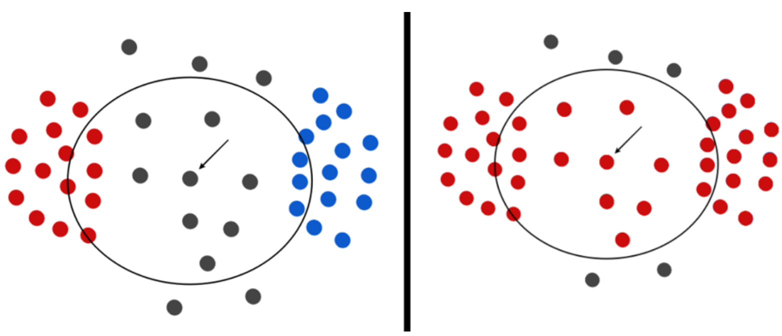

Figure 4.4 Illustration of one step of post DBSCAN

In the figure 4.4, the algorithm is visualized. Grey points are outliers, red and blue points are different clusters coming from a DBSCAN result. The circle has epsilon radius and covers the neighbors of the point that shown by arrow. After one step, all of outliers assigned as red. Then, all blue points are assigned as red as well.

Algorithm 2. The Post DBSCAN Algorithm

Input: PCD, clusterLabels: Labels from former DBSCAN result (-1 for outliers), ε: epsilon for post DBSCAN

Output: Labelled PCD

```
function postDBSCAN(PCD, clusterLabels, ε):
    for each point1 in PCD do
            clusterNumberSet ← Unique clusterLabels in ε distance to point1
            if (clusterNumberSet – {-1}) is not empty then
                smallestLabel ← min(clusterNumberSet – {-1})
                for each point2 in ε distance to point1 do
                    if label of point2 = -1 then
                        label of point2 ← smallestLabel
                    end if
            end for
            for each label1 in (clusterNumberSet – {smallestLabel}) do
                for each label2 in PCD do
                    if label2 = label1 then
                        label2 ← smallestLabel
                    end if
                end for
            end for
        end if
    end for
return PCD
```

5 THE ELEVATION REFERENCE CONNECTED COMPONENT LABELLING BASED METHOD

This method based on ER-CCL (Tian et al., 2020) algorithm. To apply proposed approach to this algorithm, 3 matrix layers are used in addition to the base method: A 75x50 binary ground matrix for separation of ground and non-ground objects, a 300x200 binary object matrix for ground object results and a 150x100 binary object matrix for non-ground segmentation.

5.1 Clipping

Clipping is needed for getting bird's-eye view from point cloud data. Because 3D LiDAR coordinates are different from 2D image coordinates. While image coordinates start from top left of the vision, LiDAR coordinates start from center and can take positive and negative values. 75 meters front, behind, and 50 meters left and right from LiDAR camera is considered, other points are removed. So, image coordinates can start 50 meters left and 75 meters front from camera and can include 150x100 meters^2 area.

5.2 Getting Bird's Eye Views

After point cloud clipping, a bird's eye views are created with 1500x1000 dimensions. Every 0,1 meter corresponds to one cell. For ground bird's eye view, ground points are 1, other points are 0. Similarly, for object bird's eye view objects points are 1, ground points are 0. To detect ground, the threshold extracting method from 4.1 session is used.

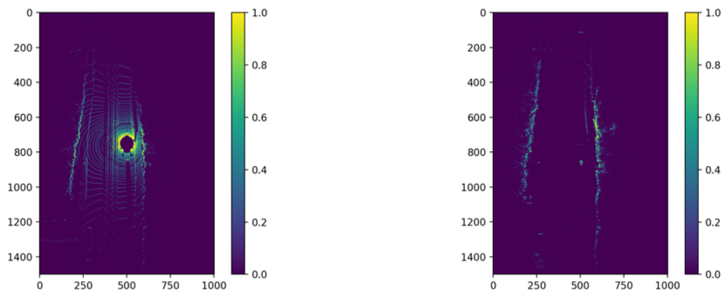

Figure 5.1: Bird's eye views for ground and objects

5.3 Down Sampling

This step should be done to execute ER-CCL algorithm and getting necessary maps. Because every point in the bird’s eye view can be not connected even if they belong to same object. Matrix cells are grouped according to down sampling rate. If there are at least one non-zero cell among matrix cell group, 1 is assigned to new cell in the down sampled object map. This operation is very similar to maximum pooling on CNN’s. For ground separation map, 75x50 matrix is obtained from ground bird’s eye view (2 meters side for each cell). Then recursive ground cleaning algorithm is executed from 4.2.1 session. Two different object maps are derived from object bird’s eye view. First one 300x200 matrix with 0.5 meter for each cell, second one is 150x100 matrix with 1 meter for each cell. First one will be used for ground object segmentation, second one is for non-ground object segmentation.

5.4 ER-CCL

This algorithm can do object segmentation by specifying and labelling connected cells. For initialization, beginning from top-left increasing integers are assigned to non-zero cells on object flag map. Then the connected cells should be defined. The surrounding 8 cells (up, down, left, right and diagonal neighbors) are selected as connected cells. After this preparation recursive algorithm is run for all cells beginning from top-left.

Algorithm 3. Recursive ER-CCL Algorithm

Input: matrix, i: row index, j: column index

Output: Labelled matrix

```
global variable adjacencyIndexes ← [ [1, -1], [1, 0], [1, 1] [-1, -1], [-1, 0], [-1, 1],
[0, -1], [0, 1] ]
function ER-CCL(matrix, i, j):
        if matrix[i][j] ≠ 0 then
                for each adjIndex in adjacencyIndexes do
                        x ← i + adjIndex[0]
                        y ← j + adjIndex[1]
                        if x < 0 or x ≥ matrix.rowCount or
            y < 0 or y ≥ matrix.columnCount then
                                continue
                        end if
                if matrix[x][y] ≠ 0 and matrix[x][y] > matrix[i][j] then
                        matrix[x][y] ← matrix [i][j]
                        ER-CCL(matrix, x, y)
                end if
        end for
    end if
```

0	1	1	1	0		0	1	2	3	0		0	1	1	1	0
0	0	1	0	0		0	0	7	0	0		0	0	1	0	0
0	1	0	0	0	Initialization	0	11	0	0	0	ER-CCL	0	1	0	0	0
0	0	0	1	1		0	0	0	18	19		0	0	0	18	18
1	0	0	1	0		20	0	0	23	0		20	0	0	18	0

Figure 5.2: The ER-CCL process on 5x5 matrix

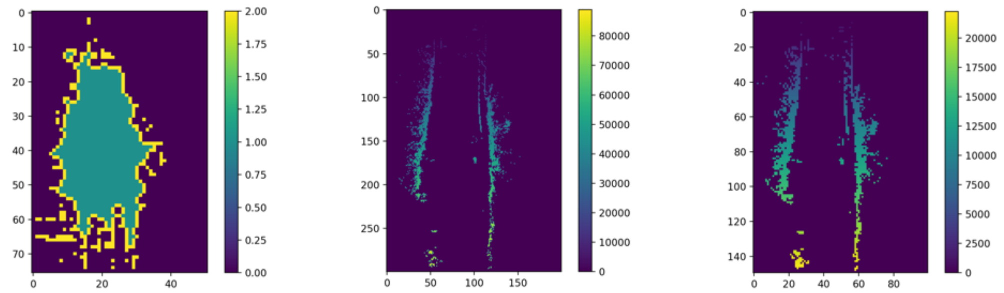

Figure 5.3: 75x50 Ground map, 300x200 object map and 150x100 object map

5.5 Result Merging

To get ultimate results, 300x200 and 150x100 object maps are merged according to ground map. If a label from 150x100 map corresponds to a non-ground area in ground map, all cells of this label is used for ultimate result. The cells are up sampled and replaced with other labels in the 300x200 matrix. Lastly, merged matrix is projected to point cloud with its labels.

6 EXPERIMENTAL STUDY

The methods are used upon some of PCD scenes from KITTI dataset (Geiger et al., 2013). The KITTI dataset is gathered from a car equipped with LiDAR, video cameras and navigation system. So, it includes mobile LiDAR data from a 360° velodyne LiDAR, high quality images from camera system and GPU/IMU data from navigation system. In addition, many benchmarks are proposed such as object detection, segmentation and tracking by labelling data and giving performance metrics. The used machine has 16 GB 3733 MHz RAM and 2.00 GHz quad-core 10th generation Intel core i5 CPU. For DBSCAN based method, execution time defends on how much non-ground points are included. It varies 5 seconds to 160 seconds among 20 scenes. For ER-CCL based method mean execution time is approximately 8 seconds. Since the ground truth values are not available, PCD is manually labelled by benefitting from DBSCAN result. A DBSCAN algorithm using small epsilon value gives too many clusters. Some of these clusters belong to same object. Such objects are detected and merged manually by visualizing DBSCAN results. By doing this, a close result to ground truth is gotten but it still includes some outlier points. The effect of the outliers going to be explained.

After getting ground truth for 20 different scenes, below metrics are computed for DBSCAN, DBSCAN based method, ER-CCL and ER-CCL based method. Then comparisons are made to see the contribution of this paper’s approach.

6.1 Clustering Performance Metrics

The used clustering metrics are defined by Rosenberg and Hirschberg (2007). They are used when real clusters are known. All of metrics can get a value between 0 and 1.

6.1.1 Homogeneity

This metric means that estimated clusters contain members of only one real cluster. The drawback of this measure is that when a real cluster separated into too many clusters homogeneity takes high values. The homogeneity equation is given below:

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{1}$$

The H(C|K) is conditional entropy between real clusters and estimated clusters.

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|} \frac{a_{ck}}{N} . \log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}} \tag{2}$$

Lastly H(C) is entropy of real clusters.

$$H(C) = -\sum_{c=1}^{|C|} \frac{\sum_{K=1}^{|K|} a_{ck}}{N} . \log \frac{\sum_{K=1}^{|K|} a_{ck}}{N} \tag{3}$$

Where N is total number of samples, a_{ij} is number of points belong to real cluster c_i and estimated cluster k_j.

6.1.2 Completeness

Completeness is the opposite metric of homogeneity. It gets high values when real clusters contain only one estimated cluster. The disadvantage is that although estimated clusters contain more than one cluster, completeness remains high. The equations are symmetrical with homogeneity:

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{4}$$

Where

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N}.\log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}} \tag{5}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{N}.\log\frac{\sum_{c=1}^{|C|}a_{ck}}{N} \tag{6}$$

6.1.3 V-measure

V-measure is harmonic mean of homogeneity and completeness:

$$v = 2.\frac{h.c}{h+c} \tag{7}$$

V-measure is the fair way to evaluate clustering, it only gets 1 if every point is clustered accurately. It is same with another metric named normalized mutual information (NMI) (Vinh et al., 2010).

6.2 Results for DBSCAN Based Method

In the table 6.1, homogeneity, completeness and v-measure results are listed for DBSCAN and our method for 20 different LiDAR scenes. Additionally, our method also has assigned outlier ratio. The DBSCAN algorithm used 1 meter epsilon value.

Table 6.1: Comparison between DBSCAN and our method in 20 scenes

DBSCAN			DBSCAN based method			
Homogeneity	Completeness	V-measure	Homogeneity	Completeness	V-measure	Assigned outlier ratio
1	0,938	0,968	0,985	0,958	0,971	0,008
1	0,856	0,925	0,941	0,933	0,937	0,022
1	0,915	0,955	0,945	0,952	0,949	0,02
1	0,808	0,894	0,911	0,87	0,89	0,031
1	0,674	0,805	0,910	0,915	0,913	0,029
1	0,898	0,946	0,934	0,966	0,95	0,016
1	0,805	0,892	0,948	0,875	0,91	0,01
1	0,823	0,903	0,9	0,906	0,903	0,027
1	0,788	0,881	0,919	0,923	0,921	0,011
1	0,86	0,925	0,962	0,957	0,96	0,011
1	0,798	0,888	0,934	0,884	0,908	0,312
1	0,796	0,886	0,959	0,915	0,936	0,019
1	0,751	0,858	0,811	0,938	0,87	0,008
1	0,675	0,806	0,89	0,963	0,925	0,023
1	0,664	0,798	0,886	0,917	0,902	0,036
1	0,728	0,843	0,978	0,975	0,976	0,007
1	0,814	0,897	0,927	0,907	0,916	0,042
1	0,793	0,884	0,974	0,964	0,969	0,019
1	0,732	0,845	0,87	0,912	0,89	0,031
1	0,797	0,887	0,951	0,913	0,932	0,029

Homogeneity of DBSCAN is always 1 since ground truth is derived from DBSCAN by merging separated clusters. So, any of clusters cannot include more than one real cluster. For our method, average homogeneity is 0,927.

The main contribution of our work can be observed by completeness. While mean homogeneity is decreased by 0,073, mean completeness is increased by 0,132 after our method. Because it merges separated non-ground objects in post DBSCAN part.

Finally, although outliers affected results negatively, mean v-measure is increased by 0,042 by our method which means more accurate overall clustering.

The Figure 6.1 shows clustering results of DBSCAN on the left and our method on the right. The clusters are shown by bounding boxes. It is clearly seen that our method has better clustering on buildings and vegetations.

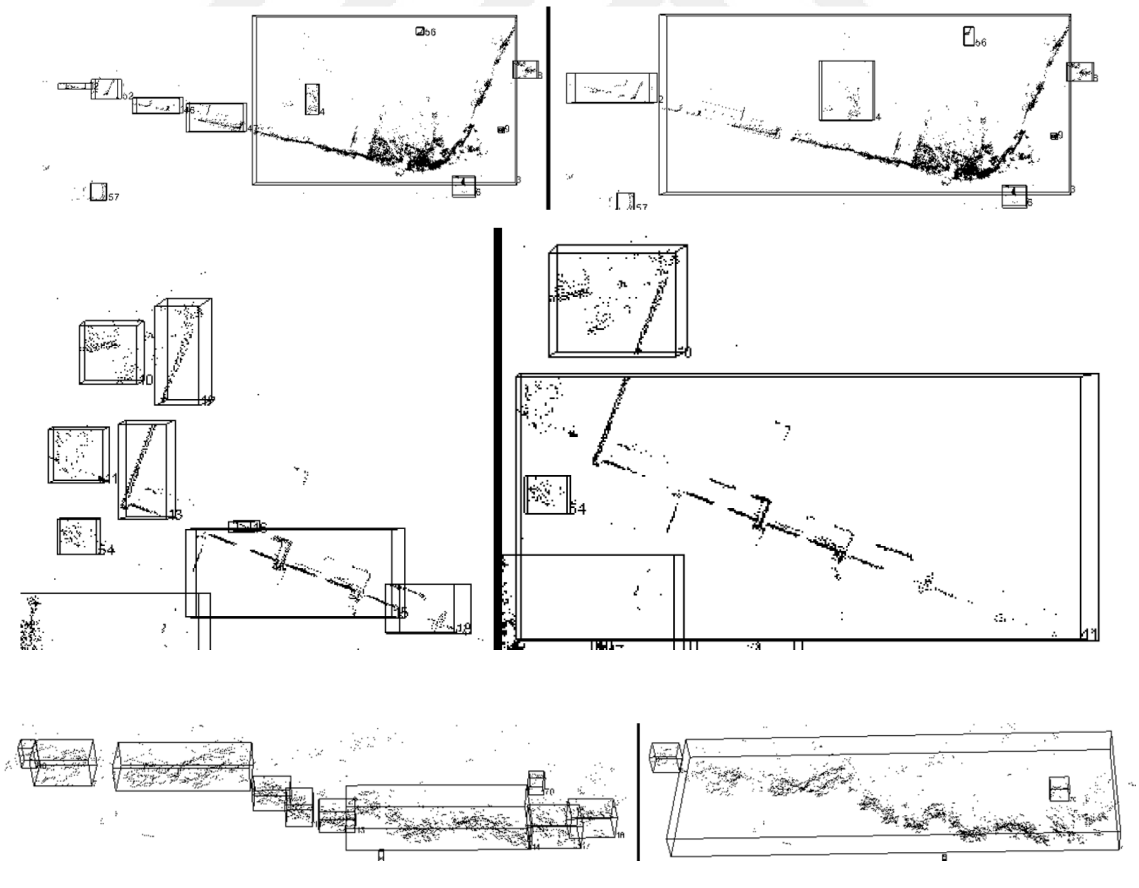

Figure 6.1 Results of DBSCAN (left) and our method (right)

6.2.1 Outlier Effect on Results

DBSCAN do clustering with outliers and gives -1 label to them. So, metrics considers outliers as a different cluster with -1 id. This causes higher completeness values for DBSCAN, lower homogeneity and completeness for our method. The assigned outlier ratio column in table 6.1 is added to estimate how much metrics are changed. The ratio is computed as below:

$$assigned\ outlier\ ratio = \frac{O_{DBSCAN} - O_{Our\ method}}{N} \tag{8}$$

Where O_{DBSCAN} is number of outliers of DBSCAN method, $O_{our\ method}$ is number of outliers of our method and N is number of total points considered in clustering.

If there was a real ground truth, all of points would belong to a real cluster, there would not be outliers. So, completeness would decrease for both methods since some of objects labelled by both a cluster id and outlier id. However, our method includes less outliers due to using higher epsilon. So, completeness of our method is less affected by outliers. On the other hand, the homogeneity of our method is decreased on the table 1 when outliers are assigned to a cluster. Because metrics consider clusters as if they include more than one cluster points. The completeness of our method is also decreased since outlier points are separated into many clusters.

In the figure 6.2, first image is DBSCAN result, second image is accepted as ground truth. Since it is obtained by merging clusters in DBSCAN, outliers are excluded which are shown by red arrows. Our method included the outliers on third image.

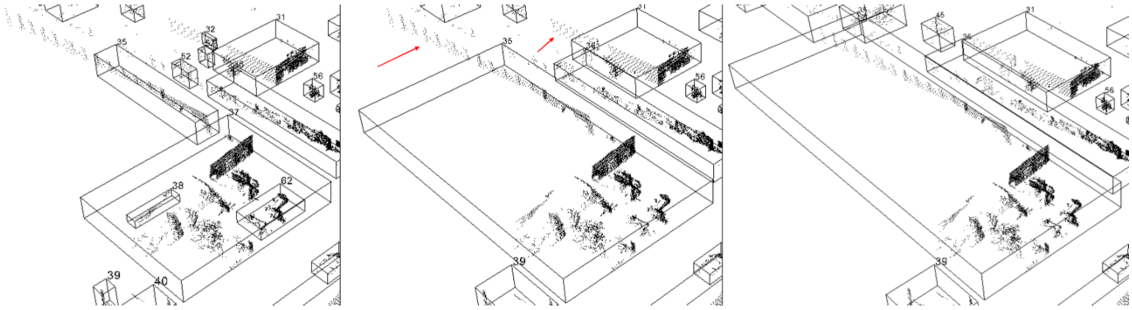

Figure 6.2 The outlier problem while getting ground truth

Eventually, although our method assigns some of outliers to clusters and gives better segmentation, this negatively affects to metrics. So, results would be better than the table with real ground truth.

6.3 Results for ER-CCL Based Method

Same metrics are computed for ER-CCL algorithm with 0.5-meter cell size and proposed ER-CCL based Method in table 6.2.

Table 6.2: Results for ER-CCL and our method

ER-CCL			ER-CCL based method		
Homogeneity	Completeness	V-measure	Homogeneity	Completeness	V-measure
0,934	0,695	0,797	0,952	0,863	0,906
0,88	0,624	0,73	0,89	0,876	0,883
0,923	0,749	0,827	0,944	0,899	0,921
0,892	0,623	0,734	0,865	0,749	0,803
0,846	0,417	0,559	0,875	0,749	0,807
0,849	0,619	0,716	0,843	0,742	0,807
0,882	0,442	0,589	0,832	0,566	0,673
0,832	0,582	0,685	0,823	0,834	0,829
0,903	0,633	0,744	0,912	0,83	0,869
0,873	0,641	0,739	0,931	0,819	0,872
0,879	0,613	0,722	0,887	0,71	0,789

0,915	0,649	0,76	0,885	0,836	0,86
0,869	0,422	0,568	0,894	0,662	0,761
0,913	0,534	0,674	0,905	0,792	0,845
0,888	0,478	0,621	0,859	0,821	0,84
0,848	0,47	0,605	0,918	0,668	0,773
0,861	0,47	0,608	0,843	0,694	0,761
0,896	0,679	0,773	0,926	0,818	0,869
0,867	0,527	0,656	0,905	0,708	0,794
0,914	0,675	0,776	0,924	0,856	0,889

According to results, mean completeness is increased by 20% and mean v-measure is increased by 13% after our approach. In terms of homogeneity, there is almost no difference because outliers from ground truth equally effects on results unlike DBSCAN based method. Since ER-CCL clusters objects from bird's eye view, there is an information lose depending on sampling rate. So, results are not as good as DBSCAN. But after our approach, metrics became closer to DBSCAN based method and ER-CCL is faster than DBSCAN. Unlike DBSCAN, this method does not have an outlier detection. Therefore, outlier objects should be excluded further. In summary, since there is information lose because of dimensionally reduction, our approach gave more contribution to ER-CLL than DBSCAN. Figure 6.3 shows ER-CCL result example above and ER-CCL based method result below. As seen, over clustering is widely prevented.

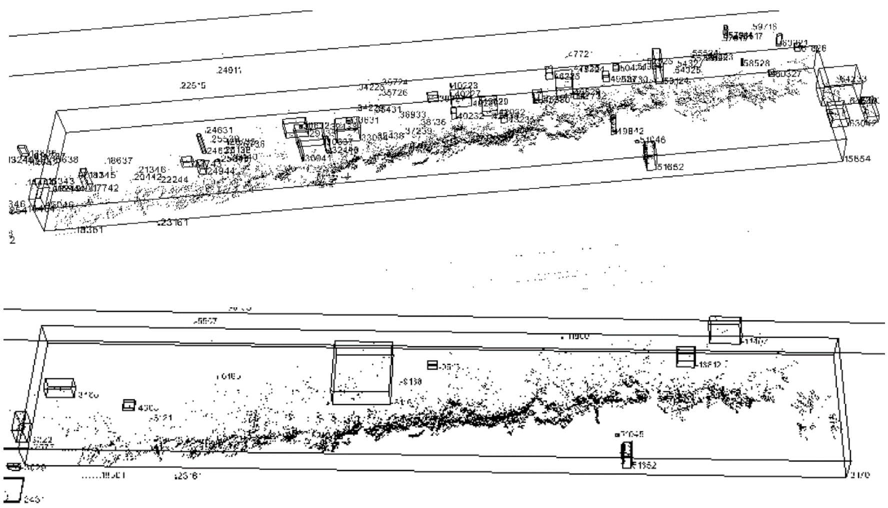

Figure 6.3: ER-CCL and ER-CCL based method on same object

7 CONCLUSION

In this thesis, an approach is proposed for object segmentation on outdoor LiDAR scenes. The approach is giving less sensitivity to non-ground objects since there are bigger objects. It is applied upon two different methods: first one is based on DBSCAN, second one is based on ER-CCL algorithm. In the DBSCAN based method, ground and non-ground objects are separated then 1 meter epsilon is applied to ground objects and 2 meters epsilon for non-ground objects. In the ER-CCL based method, 0.5-meter cell sided matrix and 1 meter cell sized matrix are merged by using another matrix that includes ground and non-ground separation.

The results are tested on different scenes and methods are compared by their base methods. While doing this comparison a close result to ground truth is obtained manually. The experiments show that proposed algorithm in DBSCAN based method gave 4% more than DBSCAN on v- measure clustering metric despite of negative effect of outliers in artificial ground truth. Because over clustering is resolved by the proposed method which can be also seen from 13% increment on completeness metric. Additionally, post DBSCAN algorithm is proposed. It merges clusters with bigger epsilon utilizing from a DBSCAN results having smaller epsilon. The ER-CCL based method provided 13% v-measure and 20% completeness refinement to its base method ER-CCL. So, proposed approach has given more contribution in CCL based method.

7.1 Future Work

To get better segmentation results from PCD, LiDAR data drawbacks should be considered. In wide ranges, the distances between points are increased just as point density. Therefore, the epsilon parameter of DBSCAN can be increased by the distance of points from LiDAR. Besides, minimum point parameter can also be arranged by distance.

If more useful data is added to method, results would be better. This study only uses x, y, z values of a point. The intensity value which is changed by material of the object can be considered. If sequential scenes are used moving and unmoving point detections would make object segmentation easier. GPS data can be beneficial in this motion detection task. In addition to LiDAR, other tools can be used to get an information fusion. The color can be used coming from a camera. Camera can also compensate point density problem of LiDAR.

In this study, the proposed approach is applied to two different attributes-based methods in literature review section. It can be used upon other types of methods. For other attributes-based methods, since they rely on clustering algorithms, parameters can be tuned for ground and non-ground points separately. The region-based methods should tend to grow regions for non-ground points rather than ground points. The graph-based methods should tend to cut edges for ground points rather than non-ground points. The deep learning methods should take ground and non-ground labels to their input.

REFERENCES

Arya, S., & Mount, D. M. (1993, January). Approximate Nearest Neighbor Queries in Fixed Dimensions. In SODA (Vol. 93, pp. 271-280).

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509-517.

Besl, P. J., & Jain, R. C. (1988). Segmentation through variable-order surface fitting. IEEE Transactions on pattern analysis and machine intelligence, 10(2), 167-192.

Bhanu, B., Lee, S., Ho, C. C., & Henderson, T. (1986, October). Range data processing: Representation of surfaces by edges. In Proceedings of the eighth international conference on pattern recognition (pp. 236-238). IEEE Computer Society Press.

Biosca, J. M., & Lerma, J. L. (2008). Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS Journal of Photogrammetry and Remote Sensing, 63(1), 84-98.

Chu, P. M., Cho, S., Park, J., Fong, S., & Cho, K. (2019). Enhanced ground segmentation method for Lidar point clouds in human-centric autonomous robot systems. Human-centric Computing and Information Sciences, 9(1), 17.

Cross, G. R., & Jain, A. K. (1983). Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, (1), 25-39.

Czerniawski, T., Sankaran, B., Nahangi, M., Haas, C., & Leite, F. (2018). 6D DBSCAN-based segmentation of building point clouds for planar object classification. Automation in Construction, 88, 44-58.

Diebel, J., & Thrun, S. (2005). An application of markov random fields to range sensing. Advances in neural information processing systems, 18, 291-298.

Dorninger, P., & Nothegger, C. (2007). 3D segmentation of unstructured point clouds for building modelling. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 35(3/W49A), 191-196.

Douillard, B., Underwood, J., Vlaskine, V., Quadros, A., & Singh, S. (2014). A pipeline for the segmentation and classification of 3D point clouds. In *Experimental robotics* (pp. 585-600). Springer, Berlin, Heidelberg

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. International journal of computer vision, 59(2), 167-181.

Filin, S. (2002). Surface clustering from airborne laser scanning data. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/A), 119-124.

Filin, S., & Pfeifer, N. (2006). Segmentation of airborne laser scanning data using a slope adaptive neighborhood. ISPRS journal of Photogrammetry and Remote Sensing, 60(2), 71-80.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11), 1231-1237.

Gelfand, N., & Guibas, L. J. (2004, July). Shape segmentation using local slippage analysis. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing (pp. 214-223).

Golovinskiy, A., & Funkhouser, T. (2009, September). Min-cut based segmentation of point clouds. In 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops (pp. 39-46). IEEE.

Hackel, T., Wegner, J. D., & Schindler, K. (2017). Joint classification and contour extraction of large 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *130*, 231-245.

Hough, P. V. (1962). U.S. Patent No. 3,069,654. Washington, DC: U.S. Patent and Trademark Office.

Huang, J., & You, S. (2016, December). Point cloud labeling using 3d convolutional neural network. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 2670-2675). IEEE.

Hubert, M., Rousseeuw, P. J., & Vanden Branden, K. (2005). ROBPCA: a new approach to robust principal component analysis. Technometrics, 47(1), 64-79.

Jiang, X. Y., Meier, U., & Bunke, H. (1996, December). Fast range image segmentation using high-level segmentation primitives. In Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96 (pp. 83-88). IEEE.

Koster, K., & Spann, M. (2000). MIR: An approach to robust clustering-application to range image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(5), 430-444.

Krishnapuram, R., & Keller, J. M. (1996). The possibilistic c-means algorithm: insights and recommendations. IEEE transactions on Fuzzy Systems, 4(3), 385-393.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., & Mitra, N. J. (2011). Globfit: Consistently fitting primitives by discovering global relations. In ACM SIGGRAPH 2011 papers (pp. 1-12).

Narksri, P., Takeuchi, E., Ninomiya, Y., Morales, Y., Akai, N., & Kawaguchi, N. (2018, November). A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles. In 2018 21st International Conference on intelligent transportation systems (ITSC) (pp. 497-504). IEEE.

Nguyen, A., & Le, B. (2013, November). 3D point cloud segmentation: A survey. In 2013 6th IEEE conference on robotics, automation and mechatronics (RAM) (pp. 225-230). IEEE.

Ning, X., Zhang, X., Wang, Y., & Jaeger, M. (2009, December). Segmentation of architecture shape information from 3D point cloud. In Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry (pp. 127-132).

Nurunnabi, A., Belton, D., & West, G. (2012, December). Robust segmentation in laser scanning 3D point cloud data. In 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA) (pp. 1-8). IEEE.

Peizhuang, W. (1983). Pattern recognition with fuzzy objective function algorithms (James C. Bezdek). SIAM Review, 25(3), 442.

Pu, S., & Vosselman, G. (2006). Automatic extraction of building features from terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(5), 25-27.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017a). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems* (pp. 5099-5108).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017b). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 652-660).

Rosenberg, A., & Hirschberg, J. (2007, June). V-measure: A conditional entropy-based external cluster evaluation measure. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL) (pp. 410-420).

Rusu, R. B., Blodow, N., & Beetz, M. (2009a). Fast point feature histograms (FPFH) for 3D registration. In 2009 IEEE international conference on robotics and automation (pp. 3212-3217). IEEE.

Rusu, R. B., Holzbach, A., Blodow, N., & Beetz, M. (2009b). Fast geometric point labeling using conditional random fields. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 7-12). IEEE.

Sappa, A. D., & Devy, M. (2001, May). Fast range image segmentation by an edge detection strategy. In Proceedings Third International Conference on 3-D Digital Imaging and Modeling (pp. 292-299). IEEE.

Schnabel, R., Wahl, R., & Klein, R. (2007, June). Efficient RANSAC for point-cloud shape detection. In Computer graphics forum (Vol. 26, No. 2, pp. 214-226). Oxford, UK: Blackwell Publishing Ltd.

Schoenberg, J. R., Nathan, A., & Campbell, M. (2010, October). Segmentation of dense range information in complex urban scenes. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2033-2038). IEEE.

Strom, J., Richardson, A., & Olson, E. (2010, October). Graph-based segmentation for colored 3D laser point clouds. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2131-2136). IEEE.

Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007, September). Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data.

Tian, Y., Song, W., Chen, L., Sung, Y., Kwak, J., & Sun, S. (2020). A Fast Spatial Clustering Method for Sparse LiDAR Point Clouds Using GPU Programming. Sensors, 20(8), 2309.

Velas, M., Spanel, M., Hradis, M., & Herout, A. (2018, April). Cnn for very fast ground segmentation in velodyne lidar data. In 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) (pp. 97-103). IEEE.

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. The Journal of Machine Learning Research, 11, 2837-2854.

Vosselman, G., & Dijkman, S. (2001). 3D building model reconstruction from point clouds and ground plans. International archives of photogrammetry remote sensing and spatial information sciences, 34(3/W4), 37-44.

Vosselman, G., Gorte, B. G., Sithole, G., & Rabbani, T. (2004). Recognising structure in laser scanner point clouds. International archives of photogrammetry, remote sensing and spatial information sciences, 46(8), 33-38.

Wang, H., Wang, B., Liu, B., Meng, X., & Yang, G. (2017). Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. *Robotics and Autonomous Systems*, *88*, 71-78.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019a). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, *38*(5), 146.

Wang, C., Ji, M., Wang, J., Wen, W., Li, T., & Sun, Y. (2019b). An improved DBSCAN method for LiDAR data segmentation with automatic Eps estimation. Sensors, 19(1), 172.

BIOGRAPHICAL SKETCH

Musa Şervan ŞAHİN has graduated from Prof. Dr. İlhan Kılıçözlü Science High School. Then, he completed computer engineering bachelor's degree at İstanbul Technical University in 2017. He worked at ING global software development department as responsible for mobile banking applications more than 3 years. He enrolled Galatasaray University Computer Engineering Master program in 2018. Currently, he is working as research engineer at Huawei R&D Center artificial intelligence enablement department.