Real-Time Plane Segmentation for Scene Understanding in Robot Navigation

Zhu Rui
Lab of Virtual Reality Technology and Systems
Beihang University, Beijing 100191, P.R. China
zhurui@buaa.edu.cn

Zhao Yongjia
Lab of Virtual Reality Technology and Systems
Beihang University, Beijing 100191, P.R. China
zhaoyongjia@buaa.edu.cn

Abstract—**Plane segmentation is one of the most fundamental procedures for scene understanding in robot navigation. However, the methods based on curvature and other higher level derivations often lead to over segmentation when used to segment point cloud with high noise and outliers. To overcome these problems, we proposed a real-time plane segmentation algorithm based on region growing which can be used to segment different planes for scene understanding in indoor environment. The proposed method contains three stages: data processing, normal estimation and region growing. In the data processing step, a pass-through filter was used to remove unnecessary points. When estimating the normal for each point, the proposed method uses the combination of octree and KD-tree search method to search the k nearest neighbors of each point. This search method can also be used in region growing to select neighboring points of the current seed which makes the proposed method more efficient. To keep balance between efficiency and over-segmentation, we presented a segment smoothness parameter according to the points curvature. What's more, the proposed method can also detect different obstacles in the scene at the same time. The experiment using the public datasets show that the proposed algorithm can be used in robot navigation for fast planes segmentation and obstacles detection , while achieving precision, recall and accuracy as much as 99.88%.**

Keywords: scene understanding, plane segmentation, region,growing, point cloud, obstacles detection

I. INTRODUCTION

Scene understanding is one of the most important step for computer vision with many applications, such as autonomous driving, indoor and outdoor mapping and visual SLAM. The scene understanding is the highest level of visual analysis which is the further processing and analysis based on image series. As we all know, the identification and understanding of the indoor scenes is the key to process intelligent information [1]. Plane segmentation is the basic step for scene understanding with many applications, for example robot navigation, object tracking and detection, 3D reconstruction [2]. Furthermore, as a basic step of the scene understanding system, plane segmentation and objects should be processed in real-time..

Thanks to three-dimensional(3D) sensor, we can acquire data about the surrounding environment conveniently. Recently,depth cameras has developed a lot in the computer vision filed which provide us both depth images and RGB images. These cameras can be divided into two categories according to the principle:Structured Light and Time-of-flight. Cameras based on Structured Light to measure the pixels such as Kinect v1、Project Tango 1 and Intel RealSense.We can get the color images and depth images from the provided pictures [3],then to calculate the 3D camera coordinates in the pixel.At last,we can use the 3D information to generate the point cloud[4]. As a fundamental step of the scene understanding system, plane segmentation must be processed efficiently and accurately. However, the method based on geometrical derivatives such as normal and distance often lead to over-segmentation even failure when used to segmentation point clouds with high noise and outliers. Besides, the exists region growing methods are not efficient enough when applied to robot navigation. For example, Popping [5] segment planes based on plane fitting and polygonization about 2.7s for 640×480 points[6].

Noting these weakness, we proposed a real-time plane segmentation algorithm based on region-growing. Firstly, a pass-through filter was used to remove any "noise" points and a voxel grid filter can be used to down sample the point cloud to a much more manageable size. Then, the combination of octree and KD-tree search method was used to search the k nearest neighbors of each points to estimate the normal which makes the proposed algorithm more efficient. At last, the region grows from the minimum curvature and uses the combination search method to select neighboring points of the current seed point. To keep balance between efficiency and over-segmentation, we presented the segment smoothness parameter according to the points curvatures. The experimental result shows that our method has many advantages compared with other methods for the accuracy and computation time.

II. RELATED WORK

There are many plane segmentation methods have been used in computer vision recently. The region growing method is to merge the pixels that are similar enough to the same cluster. It was first introduced by Besl and Jain .At first,it was used in images processing and was then improved for 3D information segmentation. Gore[7] used TIN-structed models for region growing segmentation and the TIN was used as the seed surface. To find the good seed points, Vieira and Shimada[8] presented a new parameter, a curvature threshold, to remove the points along sharp edge. Then, a median filter was used to decrease noise and select the point seed from the remaining points. Popping[9] presented a new method based on previous work. They grow the seed regions by adding the points whose distances to the growing region less than a threshold.

978-1-5386-1937-7/17/$31.00 ©2017 IEEE

Another method based on octree and voxel grid was proposed by Woo[10] ,they separated the point cloud into small voxel grid till the voxel grid normal less than the threshold which presents the average of the dataset vector. Then, Linh[2]preformed a FV method to automatically detect boundaries of building façades and their openings. Subsequently,Ann-Vu[11] proposed the octree-based region growing method which consists of two stages: region growing and refinement process. First, the region growing step extracts the coarse segmentation using the octree-based voxel grid. Then, the output is passed through a refinement process. This method is used to complex building reconstruction and urban objects classification.

Tovari and Pfeifer [12]introduced the normal vectors as the feature of the point cloud and adjusted the plane using the neighboring points distance. But they had to update the estimate matrix incrementally which is used to calculate the plane's parameters. Dirk [5] segment the plane by computing normal and get the average normal to remove noisy data in the point cloud. A region growing method based on smooth constraint is proposed by Rabbani[13] .It uses only local surface normal and point connectivity by using k nearest neighbors or fixed distance neighbors .To improve the accuracy, Reza[4] proposed a similar plane segmentation method using the neighbor points information in the point cloud. They applied the method in robot action planning and compared their method with the latest algorithm. They showed that their method produces better result both visual quality and efficiency.

Other region growing methods, for example, color-based segmentation presented by Qingming Zhan[14] based on the colorimetrical similarity and spatial proximity. But this method only process the point cloud with RGB color values. Similar to the color-based segmentation, Yan Wang[15] extracts the seed and region growing threshold by dimensional histogram statistics method. They use the similarity of H field of HSV to detect whether two neighboring points belonging to the same region.

While the region-growing method is widely used in plane segmentation and the method is easily implemented ,there still exists some problems [10], for example, the methods are often lead to over-segmented even failure when processing the noise data .Additionally ,the previous segmentation methods have a lot of parameters ,which makes them unable to be used in the robot navigation [13].

III. OVERVIEW OF THE METHOD

The proposed method has three stages: data processing, normal estimation and region growing. Figure 1 shows the overall of the proposed method and further explained below.

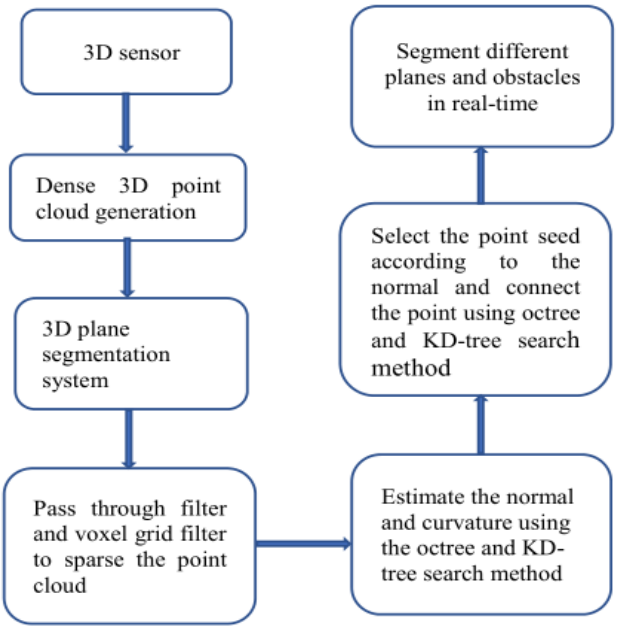


Figure1.details of the proposed method

A. *Data Processing*

Depth cameras such as Asus Xtion and Kinect sensors can provide a depth image with 307,200 points. If we process the points cloud directly ,it computation time will be very long. Because the information we use contains a lot of bad points[6].To improve the process efficiency, we perform a pass-through filter to remove unnecessary information far away from the sensors. Then, a voxel grid filter is used to down sample the point cloud to a much more manageable size.

1) Pass-through filter

A pass through filter is very simply filter that just takes each point and checks if it within a given coordinate range[16].As we all know, the more distance from the sensor, the nosier the data will be. Thus, we use a pass-through filter just discard the data far away from 2 meters.

2) Voxel grid filter

A voxel grid filter takes all the points and down samples them to a specified resolution. Point Cloud Library divides the space into small cubes and then averages the points in each cube to get a centroid[16].This can be very useful to reduce the point cloud to a much more manageable size. We set the leaf size of 1cm for 307,200 points and 1mm for 19,200 points. Figure 2 shows a data processing sample.

Figure2. Example of data processing.The original point cloud (left) and the data processing output(right)

B. *Normal estimation*

Surface normal is an important property of geometric surface.The normal for each point can be estimated in unstructured point cloud by using k nearest neighbors(KNN). Euclidean was used as the distance metric for it is efficient to process the massive points. Surface normal is the important property of the geometry objects and has a lot of applications.Moreover, to avoids light changing,the normal estimation method choose parameters according to the point cloud number[13].Search for KNN can be optimized by using different space partitioning strategies like octree, KD-tree. To eliminate the distance error and improve the segment accuracy, we use combination of octree and KD-tree search method to search the k nearest neighbor points. In this paper, we are using Normal Estimation class in the Point Cloud Library. The output surface curvature is estimated as a relationship between the eigenvalues of the covariance matrix .The experimental result shows that our proposed method is much more efficient than previous methods[5].

C. *Region growing*

The most important steps in the proposed method is region growing which uses the point normal and their curvatures to divided the points cloud into different clusters. At first , it sorts the points by their curvatures value. Then, we grow the points into different regions from the point which has the minimum curvature value. The process of region growing is shown in Algorithm 1.This method tries to group different point which has similar smooth surface together. Angle threshold and Curvature threshold are two important parameters in the algorithm ,because we use them to decide whether the points belong to the same objects surface. In our method,if the points belong to the same objects ,their normal deviation will be the same or almost the same one. At last, all the planes and obstacles are segmented in real-time. When using the region growing method to segment different scenes, we presented a segment smoothness parameter β to keep balance between over-segmented and under-segmented automatically which is defined in (1).

$$\beta=\sqrt{(pc_x\text{-}pc)^2+(pc_y\text{-}pc)^2+(pc_z\text{-}pc)^2} \quad (1)$$

$$pc=(pc1+pc2)/2 \quad (2)$$

Where pcx is the principal curvature X direction,pcy is the principal curvature Y direction and pcz is the principal curvature Z direction. pc1 presents the max eigenvalue of curvature and pc2 presents the min eigenvalue of curvature. We set the Angle threshold and Curvature threshold as the following.

$$\theta_{th}=\beta/180.0*MI\_P \quad (3)$$

$$C_{th}=\beta \quad (4)$$

Algorithm 1 region growing algorithm using normal and curvatures

Inputs: Point cloud=$\{\boldsymbol{P}\}$, point normal vectors$\{\boldsymbol{N}\}$, point curvatures=$\{\boldsymbol{C}\}$, neighbor finding function $\Omega(.)$,min-number of cluster, Curvature threshold $\mathbf{C_{th}}$, Angle threshold θ_{th}

Initialize Region List$\{R\} \leftarrow \emptyset$,Available points list$\{\mathrm{A}\} \leftarrow \{1 \ldots . P_{count}\}$

While$\{\mathbf{A}\}$ is not empty do
  Current region$\leftarrow \emptyset$,Current seeds$\{S_c\} \leftarrow \emptyset$
  Point with minimum curvature in $\{\mathrm{A}\} \leftarrow P_{min}$
  $\{\boldsymbol{S_c}\} \leftarrow \{\boldsymbol{S_c}\} \cup \boldsymbol{P}_{min}$
  $\{\boldsymbol{R_c}\} \leftarrow \{\boldsymbol{R_c}\} \cup P_{min}$
  $\{\mathbf{A}\} \leftarrow \{\mathbf{A}\} \backslash P_{min}$
  for i=0 to size ($\{S_c\}$) do
    Find nearest neighbors of current seed point $\{B_c\} \leftarrow \Omega(S_c)$
    for j=0 to size ($\{\boldsymbol{B_c}\}$) do
      Current neighbor point $P_j \leftarrow B_c\{j\}$
      if $\{\mathrm{A}\}$ contains P_j and $\cos^{-1}(|N\{S_c\{i\}\}, N\{S_c\{j\}\}|) < \theta_{th}$ **then**
        $\{\boldsymbol{R_c}\} \leftarrow \{\boldsymbol{R_c}\} \cup \boldsymbol{P}_j$
        $\{\mathrm{A}\} \leftarrow \{\mathbf{A}\} \backslash P_j$
        if $C_p < C_{th}$ **then**
          $\{\boldsymbol{S_c}\} \leftarrow \{\boldsymbol{S_c}\} \cup \boldsymbol{P}_j$
        end if
      end if
    end for
  end for
  Add current region to global segment list$\{\boldsymbol{R_c}\} \rightarrow \{\mathbf{R}\}$
end while
Return $\{\boldsymbol{R_c}\}$

Figure3. shows an example of plane segmentation using different Angle threhold.

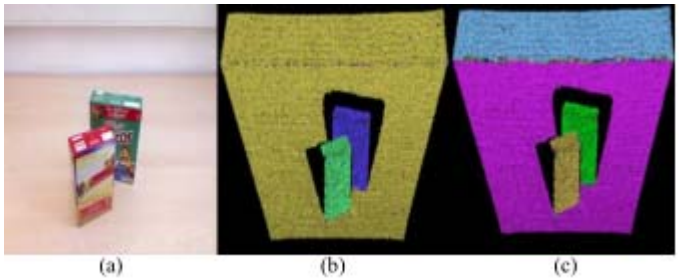


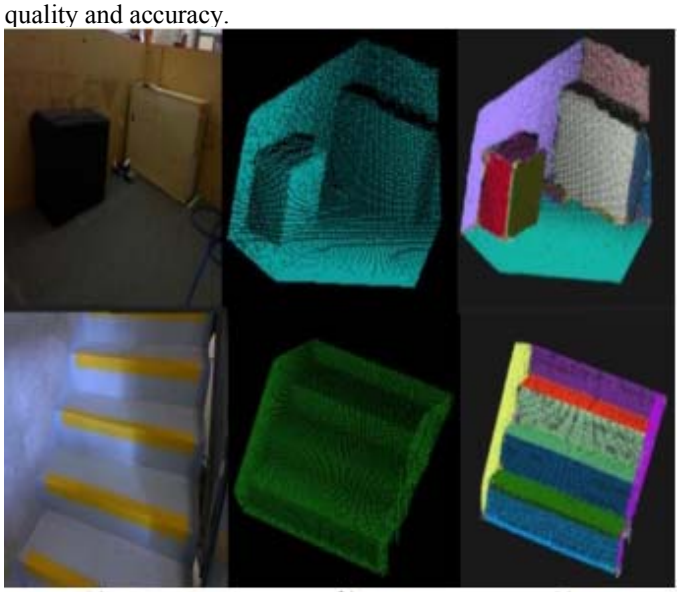
Figure4.An example of plane segmentation.(a)is the original color image.(b)is the segmentation result using θuh=10.0/180.0*M_PI.(c)shows the result using θuh =6.0/180.0*M_PI according to the proposed algorithm.

IV. EXPERIMENT TEST

To determine efficiency and accuracy of our method,we test the algorithm using two different datasets. The first dataset[4] was captured during robot rescue competitions,using ASUS Xtion PRO LIVE with 640×480 resolutions.The dataset consists of a set of common scenes, including stairs,walls, ground and boxes. The second

experiment dataset comes from OSD[17] ,including boxes, stacked boxes, cylindric objects and complex scenes. The dataset is acquired from 3D sensors recorded at full frame rate(30FPS) with 640×480 resolutions.These cameras can measure the depth of each pixel then get the color and distance information.We can calculate the 3D coordinate of each pixel and transform them into point cloud .For all images ground truth segmentations and plane parameters are available.

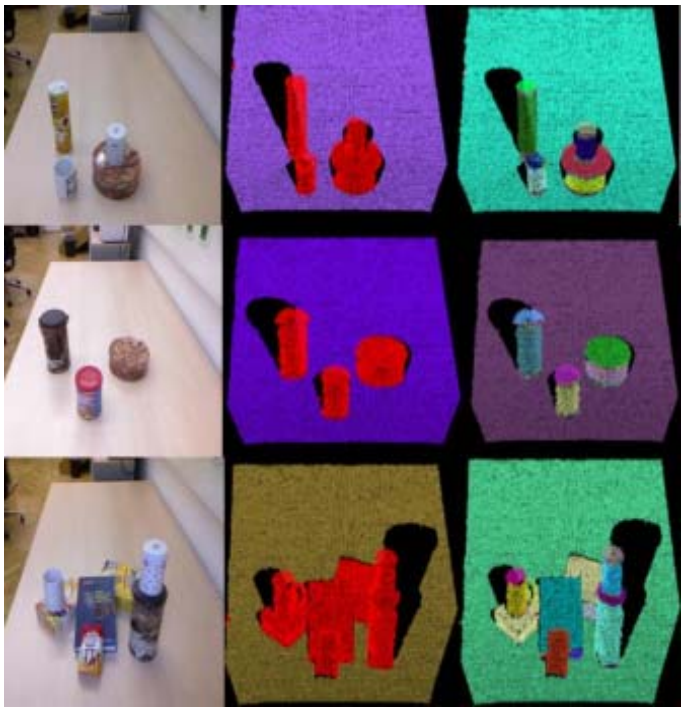
Figure 5 shows the segmentation result using the robot rescue competition dataset.As a basic step for robot navigation, it is virtual to detect different planes in real-time accurately. Different planes are represented by different colors.Result show that our method can detect different planes and objects successfully. We employed 80 as the minimum region size.According the our experiment experience,the number of the region size too big or too small both cannot get the reasonable result. Besides, we used the combination of octree and KD-tree search method to estimate the normal and find the point seed, so the segment result is quite good both visual quality and accuracy.



(a) (b) (c)

Figure5.Plane segmentation results of two scenes from robot rescue competition dataset[4].(a) is the color images of two different indoor scenes.(b) is the original point clouds.(c)shows the result of different plane segmentation

Figure 6 shows plane segmentation and obstacle segmentation from the OSD dataset. (a) is the original color images.(b) shows our segment result, and we regard all the objects as the obstacles except the desktop.At first, all the planes are extracted, then,different obstacles are distinguished using the same method according to their normal and curvatures. As we can see,the proposed method can segment the planes and obstacles in real-time, which makes it can be used in many applications such as indoor navigation system and scenes analysis system.



(a) (b) (c)

Figure 6.Plane segmentation and obstacle detection results of three scenes from OSD RGB-D dataset.(a) is the color images of three different scenes. (b) shows the result of plane segmentation and obstacles are painted with red. (c) shows the result of obstacle detection. Different obstacles are painted with different colors randomly.

V. RESULT AND DISCUSSIONU

The proposed algorithms in this paper were tested using C++ and run on an i5-6300HQ CPU @2.30 GHZ and with 4.0GB RAM. We don't use any OpenMP or GPU to effect the implement efficiency.TABLE 1 shows the experiment result using the OSD dataset. The segmentation time includes all the steps of the method. In order to evaluate the performance of the proposed method, we use the standard measures widely used for classification---precision recall[18]and accuracy. Precision recall and accuracy are defined as follows:

$$\text{precision}=\frac{TP}{TP+FP} \tag{5}$$

$$\text{recall}=\frac{TP}{TP+FN} \tag{6}$$

$$\text{accuracy}=\frac{TP}{TP+TN+FN+FP} \tag{7}$$

TP denotes the number of true positive and FP is the number of false positive.FN is false negative. They are used in many paper to determine the segment result.After we get the segment result ,we analyze the segmentation result according to the ground truth labels.

TABLE 1 shows the performance of the method using the robot rescue competition dataset and OSD dataset .The first two examples come from the robot rescue competition dataset[4]The detection time includes all the steps.

TABLE 1: EVALUTION OF THE PERFORMANCE OF THE PROPOSED

Scene name	Detection time(ms)	Precision	Recall	Accuracy
Scene1	10.67	98.10	98.12	98.07
Scene2	10.53	99.88	99.60	99.71

TABLE 2 shows performance of the plane detection and obstacle detection using the OSD dataset with 640×480 points for each scene. In order to objectively evaluate our method, we calculate the plane segmentation and obstacles detection precision, recall and accuracy, respectively.

TABLE 2: THE PERFORMANCE OF PLANE SEGMENTATION AND OBSTACLES DETECTION RESULT USING OSD DATASET

Scene name	Detec-tion time(ms)	Plane Segmentation			Obstacle detection		
		Precisi-on	Recall	Accu-racy	Precisi-on	Reca-ll	Accu-racy
Scen-e1	41.29	99.98	99.97	99.96	99.78	99.60	99.67
Scen-e2	44.40	99.99	99.79	99.98	99.80	99.65	99.68
Scen-e3	46.53	99.97	99.77	99.87	96.23	96.66	96.14

The result shows that the proposed method can be used for scene understanding in robot navigation with high precision, recall and accuracy. As we can see,the proposed method is quite efficient with an average of detection time 10.606 ms for 160×120 points and 45.07 ms for 640×480 points. Compared to Holz's method[5],we obtained better segmentation results(they report runtimes of 15.3ms for 160×120,and >129ms for 640×480). Thus, the proposed method is efficient enough to develop the real-time application for scene understanding in robot navigation. Since all the planes and obstacles can be detected at the same time,we don't need to remove the planes. Even though there are less planar area,the plane segmentation results have little effects on the performance of obstacle detection ,which makes our method very suitable for complex scenes.

VI. CONCLUSION

Segmentation is an important step for scenes understanding in robot navigation. In this paper, we proposed a real-time plane segmentation and obstacle detection algorithm based the region-growing method. Our main contribution is using the combination of octree and KD-tree searching method to keep balance between efficiency and over-segmented. What's more, we presented a segment smooth parameter to make the proposed method more accurate. The experimental result shows our method is quite efficient and accurate with an average accuracy 99.77% in simple scenes and 96.60% in complex scenes. Thus, the proposed algorithm is very suitable for scene understanding in indoor environment. In the future, we will focus on combining the color information acquired from the Kinect and ASUS Xtion PRO LIVE in order to improve the accuracy of obstacles detection.

REFERENCES

[1] Z. Wang, H. Liu, Y. Qian, and T. Xu, "Real-time plane segmentation and obstacle detection of 3D point clouds for indoor scenes," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7584 LNCS, no. PART 2, pp. 22–31,2012.

[2] L. Truong-Hong and D. F. Laefer, "Octree-based, automatic building fa??ade generation from LiDAR data," CAD Comput. Aided Des., vol. 53, no. August, pp. 46–61, 2014.

[3] S. Gachter, V. Nguyen, and R. Siegwart, "Results on range image segmentation for service robots," Comput. Vis. Syst. …, no. Icvs,2006.

[4] R. F. B, "Region-Growing Planar Segmentation," pp. 179–191, 2015.

[5] D. Holz and S. Behnke, "Approximate triangulation and region growing for efficient segmentation and smoothing of range images,"Rob. Auton. Syst., vol. 62, no. 9, pp. 1282–1293, 2014.

[6] R. A. Zeineldin and N. A. El-Fishawy, "Fast and accurate ground plane detection for the visually impaired from 3D organized point clouds," Proc. 2016 SAI Comput.Conf. SAI 2016, pp. 373–379, 2016.

[7] B. Gorte, "Segmentation of TIN-structured surface models," Int. Arch.Photogramm. Remote Sens. Spat. Inf. Sci., vol. 34, no. 4, pp. 465–469, 2002.

[8] M. Vieira and K. Shimada, "Surface mesh segmentation and smooth surface extraction through region growing," Comput. Aided Geom. Des., vol. 22, no. 8, pp. 771–792, 2005.

[9] J. Poppinga, N. Vaskevicius, A. B irk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3D range images," 2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS, pp.3378–3383, 2008.

[10] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A new segmentation method for point cloud data," Int. J. Mach. Tools Manuf., vol. 42, no.2, pp. 167–178, 2002.

[11] A. V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," ISPRS J.Photogramm. Remote Sens., vol. 104, pp. 88–100, 2015.

[12] D. Tovari and N. Pfeifer, "Segmentation based robust interpolation -- a new approach to laser data filtering," Laserscanning 2005, vol.IAPRS Vol, p. 6, 2005.

[13] T. Rabbani, F. a van den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," Int. Arch. Photogramm.Remote Sens. Spat. Inf. Sci. - Comm. V Symp. 'Image Eng. Vis.Metrol., vol. 36, no. 5, pp. 248–253, 2006.

[14] Q. Zhan, L. Yubin, and Y. Xiao, "Color-Based Segmentation of Point Clouds," Laser scanning 2009, IAPRS, vol. XXXVIII, P, pp. 248–252, 2009.

[15] C. Huang et al., "Foundations of Intelligent Systems," Adv. Intell.Syst. Comput., vol. 277, pp. 911–919, 2014.

[16] J. H. Palnick, T. Padir, and J. Palnick, "Plane Detection and Segmentation For DARPA Robotics Challange," 2014.

[17] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze,"Segmentation of unknown objects in indoor environments," IEEE Int. Conf. Intell. Robot. Syst., pp. 4791–4796, 2012.

[18] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," Proc. 23rd Int. Conf. Mach. Learn. --ICML'06, pp. 233–240, 2006.