IEEE UPINLBS'2016
Nov. 3~4, 2016, Shanghai, P.R. China

Real-Time Plane Segmentation in a ROS-based Navigation System for the Visually Impaired

Ke Jin, Peilin Liu, Rongdi Sun, Zhenqi Wei, Zunquan Zhou
School of Electronics Engineering
Shanghai Jiao Tong University
Shanghai, China

***Abstract*—This paper provides a real-time plane segmentation method which can be used in navigation systems for the visually impaired to avoid indoor obstacles. The proposed method is based on surface normal estimation in range images. Efficiency and overall accuracy are considered as two main challenges in plane segmentation algorithms which use depth information. Our method exploits integral images to enhance the efficiency of normal estimation. A dynamic determination for smoothing region is proposed in our method to improve the overall accuracy. Compared to the methods in Point Cloud Library (PCL), our method consumes less time and has better performance in a wide range of depth (1~8m). The proposed method is implemented on Robot Operating System (ROS) at 30fps. Our method makes it possible to obtain a robust and real-time indoor navigation system integrated with commercial Time-of-flight (TOF) sensors.**

Keywords—plane segmentation; normal estimation; navigation system; range image; time-of-flight

I. INTRODUCTION

This paper provides a real-time plane segmentation method which can be used in navigation systems for the visually impaired to avoid indoor obstacles. As reported by World Health Organization in 2015, the number of the visually impaired globally reached over 200 million. Such disabled people suffer great danger and hazards from various obstacles in indoor environment. White canes are often used to help the visually impaired people, however, they can only detect ground obstacles within 1m. A guide dog is another choice but the question is its high price. Recently, navigation systems based on computer vision are developed to provide robust and relatively low-cost obstacle avoidance service to enhance the visually impaired people's safety.

Plane segmentation is a critical process in domestic navigation since planar structures are abundant in indoor environment, such as the floors, the walls and the tables. The segmented planar structures can provide a quick understanding of the surrounding environment. Real-time navigation services can be conducted via analyzing the position of the segmented planes. Furthermore, plane segmentation is closely related to obstacle recognition. Huge obstacles such as desks or cupboards can be considered as a combination of one or more planes. On the other hand, scattered obstacles can be extracted after the backgrounds (often planes) are removed. Plane segmentation is therefore regarded as the pre-process of obstacle recognition. Fig. 1 shows the segmented planes which can be used in navigation systems to help the visually impaired people to avoid obstacles in domestic environment.

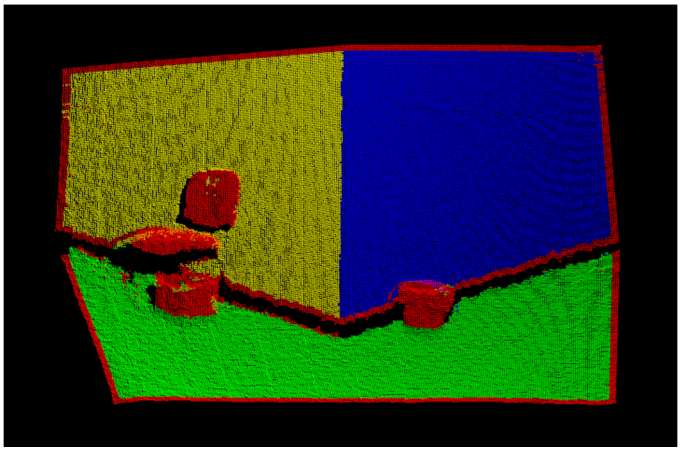


Fig. 1. An indoor scene with segmented planes and obstacles. Green planes are safe. Blue and yellow planes are giant planes which usually represent walls or roofs. The outline and obstacles are labeled in red.

Planar structures can be computed by depth information [1][2]. The recent development of depth sensors [3], like TOF sensors, can provide fast distance measurement to the observes scene. Such depth sensors can generate range images which can be converted into organized 3D point clouds in real-world coordinate. Normally, depth data from the commercial TOF sensors contain noise which influences the accuracy of plane segmentation.

Recently, plane segmentation methods can be divided into three groups: (1) edge based segmentation [4], (2) normal based segmentation [5-9], and (3) random sample consensus (RANSAC) [10]. As stated in [4], edge based segmentation has the least consumption of time but the worst performance of accuracy against noise. Normal based segmentation can be divided into two groups: averaging and optimization-based methods. In [5], normal based segmentation methods have proven more reliable but more time-consuming than edge based methods. RANSAC method can conduct quite robust plane segmentation with consuming a large amount of time.

Since these three methods have their own disadvantages to be applied in navigation systems, we propose a real-time normal based (RTNB) plane segmentation method in this paper. The proposed method computes the surface normal based on averaging algorithms. Integral images [11] are exploited to enhance the efficiency in normal estimation. In addition, a dynamic determination for smoothing region [12] and a weight

978-1-5090-2879-5/16/$31.00 ©2016 IEEE

assignment approach are proposed in our method to improve the overall accuracy.

The RTNB method is implemented on ROS at 30fps. ROS is a flexible framework for writing smart software. Since ROS is an open source project, the RTNB method can be improved in an international community. Our proposed RTNB method makes it possible to obtain a robust and real-time indoor navigation system integrated with commercial TOF sensors.

The algorithm flow of the RTNB method is described in Sec II. The detailed introduction of the key points in the method can be found in Sec III. Eventually, we compared the RTNB method to other two methods in PCL in Sec IV.

II. Algorithm Flow

The RTNB plane segmentation is based on surface normal estimation. The algorithm flow chart is shown in Fig. 2. In the main flow, the input data are range images with noise from TOF sensors. Firstly, the range images are converted into organized 3D point clouds in real-world coordinate through linear transformation. Then we computed the surface normal of each 3D points based on averaging methods. Eventually, we exploit the region growing method [13] to classify surface normal with similar orientation angles so that various planes are segmented.

Normal estimation is the key process in the RTNB plane segmentation method. The efficiency and accuracy of plane segmentation greatly depend on the normal estimating results. In the process of surface normal estimation, we take the following steps. Firstly, we choose the point of interest (POI) whose normal is going to be estimated and find the four neighboring points which are upper, lower, left and right to the POI in a certain distance. Secondly, we applied a weighted averaging filter to the four points to reduce the influence of noise. Integral images are used in this step to reduce the data access. The smoothing region of the filter is dynamically determined in our method. A weight assignment approach is also conducted in this filter. After the positions of the four points are smoothed, we compute the horizontal vector via the left and right points and the vertical vector via the upper and lower points. Eventually, the cross-product between these two orthogonal vectors are computed as the surface normal for the POI.

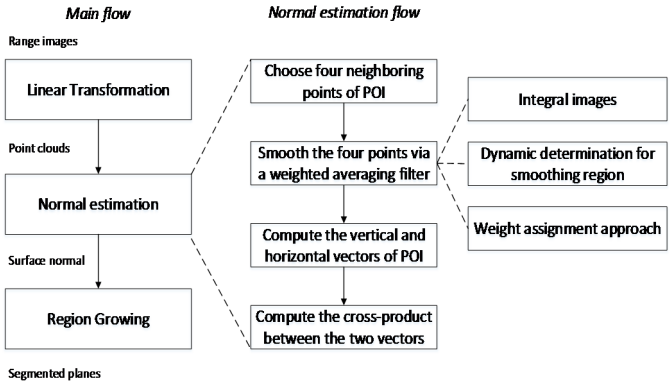


Fig. 2. Algorithm flow chart of the RTNB method.

III. Key Points In RTNB Method

In the following section, we describe the key points in the RTNB method. Firstly, the principle of integral images is introduced in Sec. III-A to show how the processing efficiency is enhanced. Secondly, a dynamic determination for smoothing region is proposed in Sec III-B. Thirdly, a weight assignment approach is introduced in Sec. III-C. These two points are aimed to improve the overall accuracy of normal estimation. Then we describe the standard way of normal estimation in Sec. III-D based on the first three points. Finally, a region growing method is introduced to classify all the points in normal space in Sec. III-E.

A. Integral Images

In the process of surface normal estimation, we need to calculate the sum of all the values within a certain region where the averaging filter is covered. That means if the smoothing region is chosen to be a $m*n$ rectangular area, we need to access $m*n$ data elements in each smoothing of one point. This large amount of data access leads to huge time consumption.

Integral images can be used to reduce the data access. The value of each pixel in an integral image I_G is the sum of all the upper and left pixels in the origin image G. Each pixel element (u,v) is defined mathematically as the sum of the rectangular area between $G(0,0)$ and $G(u,v)$ in (1):

$$I_G(u,v)=\sum_{i=0}^{u}\sum_{j=0}^{v}G(i,j) \tag{1}$$

The integral images can be computed efficiently through iteration in (2):

$$\begin{aligned} I_G(u,v) &= G(u,v) \\ &+ I_G(u-1,v) \\ &+ I_G(u,v-1) \\ &- I_G(u-1,v-1) \end{aligned} \tag{2}$$

An integral image I_G makes it possible to compute the sum of all values within a certain rectangular region R in the origin image G by accessing only four data elements at all. Any rectangular sum can be computed in four region references. Therefore, the average value in a region of interest can be calculated in (3):

$$\begin{aligned} A(I_G,u,v,d) = (1/4r^2)\cdot[\, & I_G(u+d,v+d) \\ & + I_G(u-d,v-d) \\ & - I_G(u+d,v-d) \\ & - I_G(u-d,v+d)\,] \end{aligned} \tag{3}$$

(u,v) is regarded as the center point and d is the radius of the rectangular mesh for averaging.

B. Dynamic Determintation For Smoothing Region

Range images from commercial TOF sensors contain noise which may influence the results of the normal estimation. Poor accuracy of the estimated surface normal would decrease the robustness of plane segmentation.

The accuracy of the normal estimation largely depends on the size of the smoothing region. In original works, the smoothing region are set fixed in surface normal estimation. A fixed smoothing region leads to a poor overall accuracy of normal estimation in a wide range of depth. We proposed a dynamic approach to determine the size of the smoothing region so as to enhance the overall accuracy of normal estimation.

In our research, an indicator is figured out for estimating the size of the smoothing region for the POI. The indicator is the depth of the POI, i.e. the larger the depth is, the worse signal-to-noise ratio can be achieved. In a farther distance, we need to enlarge the smoothing region to maintain the accuracy of normal estimation. In our research, we compute the surface normal on an absolutely flat plane in eight depth levels (1~8m). We take a triangular mesh as the smoothing region. In each depth level, the surface normal is estimated by various size of smoothing radius. Fig. 3 shows the result of normal estimation accuracy. The accuracy gets worse with the increase of the depth level at the same size of smoothing region. In each depth level, the accuracy becomes better with the increase of smoothing radius and it shows a saturate tendency at a certain accuracy. If we set the saturate accuracy as an acceptable accuracy of normal estimation in each depth level, we can fit a curve to show the relationship between depth and the radius of smoothing region in Fig. 4. It can be described using the function in (4):

$$d(u,v)=\alpha\cdot\{\beta-\exp[P_z(u,v)/\gamma]\} \tag{4}$$

In (4), $d(u,v)$ is the smoothing radius depending on the depth value $P_z(u,v)$ of (u,v) while α,β,γ are the constants depending on the depth sensors. In our research, we estimated $\alpha=8.98$, $\beta=1.02$, $\gamma=4.99$. By (4), we get the relationship between the size of smoothing region and the depth. Therefore, the RTNB method can provide a dynamic determination for smoothing region at each point with a certain depth.

C. Weight Assignment

The smoothing region is chosen to be a rectangular mesh with a various radius determined by the depth of the POI. However, if the size of smoothing region is only determined by the depth we would also smooth over object borders. It may cause the normal estimation error in object edges.

The problem is considered in our research and solved by employing a weight assignment approach in the averaging filter. The weights for different neighboring pixels are set by step edge detectors based on depth. We employed simple arithmetic to emulate the smoothing process of the first derivative of the depth map as shown in (5):

$$M(p)=\Sigma_{i\in W(p)}\begin{cases}1 & |P_z(i)-P_z(p)|<t\\ 0 & otherwise,\end{cases} \tag{5}$$

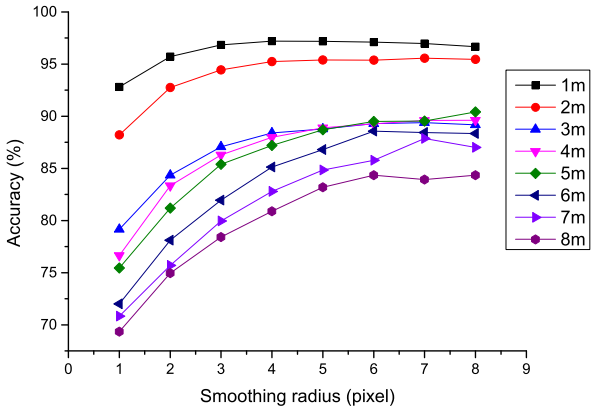


Fig. 3. The accuracy of normal estimation in eight depth levels with respect to various smoothing radius.

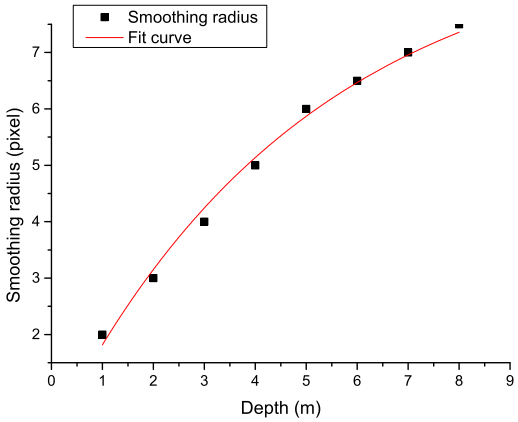


Fig. 4. A fitted curve to show the relationship between the depth and the smoothing radius for the saturate accuracy.

where the output of (5) is a gradient map $M(p)$ with pixels representing gradient of the edge, $W(p)$ is the window of neighboring pixels of the point p . The threshold t is set for specifying a step edge. Then we applied another threshold th in the gradient map to get an edge detecting map $E(p)$ with pixels representing the steepness of edge in (6):

$$E(p)=\begin{cases}M(p)-th & M(p)>th\\ 0 & otherwise.\end{cases} \tag{6}$$

The edge detecting map determines the weight for each neighboring pixels of the POI. It is shown in Fig. 5 that two conditions are considered depending on the location of the POI. If the POI is not on the edge, the neighboring pixels at the same side with the interested point has a higher weight, otherwise, they weight lower according to the steepness of the edge. If the POI is on the edge, then only a small size of region would be involved for smoothing. Pixels on the edge weight zero at both conditions.

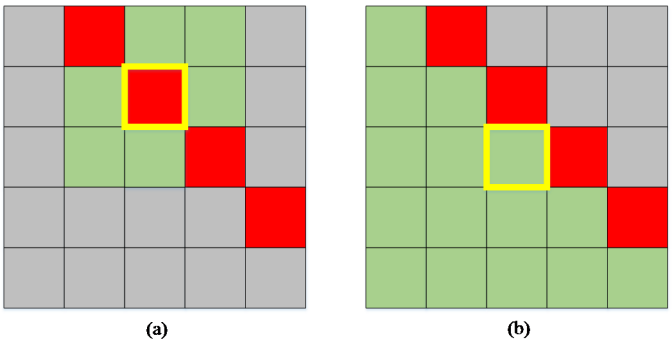


Fig. 5. Pixels with yellow frames are the points of interest. Pixels in green weight more while pixels in gray weight less. Red pixels on edges weight zero. (a) Weights for neighboring pixels if the POI is on the edge. (b) Weights for neighboring pixels if the POI is not on the edge.

D. Normal Estimation

We adopted a standard way to calculate the surface normal $n_{(u,v)}$ at a certain point (u,v). Cross-product between two orthogonal vectors based on the neighboring pixels of (u,v) are computed as the surface normal in (7):

$$n_{(u,v)} = h_{(u,v)} \times v_{(u,v)} \tag{7}$$

The horizontal vector $h_{(u,v)}$ is computed between the left and right neighbor of (u,v) while the vertical vector $v_{(u,v)}$ is computed between its upper and lower neighbor.

Based on the weighted averaging filter, we can derive the 3D vectors $h_{(u,v)}$ and $v_{(u,v)}$ by (8):

$$\begin{cases} h_{x(u,v)} = [P_x(u,v+d(u,v)) - P_x(u,v-d(u,v))]/2 \\ h_{y(u,v)} = [P_y(u,v+d(u,v)) - P_y(u,v-d(u,v))]/2 \\ h_{z(u,v)} = [A(I_{z,G},u,v+d(u,v),d(u,v)) \\ \qquad - A(I_{z,G},u,v-d(u,v),d(u,v))]/2 \\ v_{x(u,v)} = [P_x(u+d(u,v),v) - P_x(u-d(u,v),v)]/2 \\ v_{y(u,v)} = [P_y(u+d(u,v),v) - P_y(u-d(u,v),v)]/2 \\ v_{z(u,v)} = [A(I_{z,G},u+d(u,v),v,d(u,v)) \\ \qquad - A(I_{z,G},u-d(u,v),v,d(u,v))]/2 \end{cases} \tag{8}$$

P_x, P_y, P_z are the 3D data which represent the value of x-, y-, z-coordinates of the organized point clouds. Due to the noise in data, a smoothing region (a rectangular mesh) is dynamically determined to reduce the normal estimation error. The smoothing region $d(u,v)$ can be derived by (4). In addition, $I_{z,G}$ is the integral image of the z-component of the point clouds.

E. Region Grow

After the surface normal estimation is finished, all the normal needs to be classified into different groups. We employed a region growing algorithm for segmenting planes. Region Grow is a method to gradually form a complete and independent component among huge amount of pixels based on some pre-defined growing criteria. For the region of interest, a seed point means it is consistent with the criteria that we pre-defined. The algorithm starts from one of the seed points and checks the neighboring pixels whether they are also consistent with the criteria. New seed points are generated from the neighbor until all their neighboring points are against the criteria, then region growing stops. The iterative steps in our algorithm for region growing include:

- Start to extract a planar region by selecting a seed point manually;
- Check if the neighboring pixels has a similar surface normal (the angle difference should be less than a threshold);
- Set all the similar neighboring pixels as seed points and continue checking their neighbor;
- Stop region growing if no new seed points exist.

After region growing process is finished, segmented planes can be translated into warnings or guiding messages by a navigation system to help the visually impaired people at indoor environment.

IV. Results

In this section, the RTNB plane segmentation method is compared to two state-of-the-art plane segmentation methods in PCL. These two methods are the Optimization-Based plane segmentation method and the RANSAC plane segmentation method. In Sec. IV-A, we compare the RTNB method with the other two methods in the processing efficiency. In Sec. IV-B, we compare the RTNB method with the Optimization-Based method in the overall accuracy of normal estimation. All experiments are performed on a laptop with a 2.6 GHz Intel(R) Core(TM) i7-4720HQ CPU and an 8GB RAM, while all the three methods are single-thread.

A. Processing Speed

An average computational time of the three segmentation methods run on 320x240-pixel images was measured. The results of computational time are summarized in Table I. The input range images in this experiment are from a commercial TOF sensor. We take 1000 frames randomly in indoor environment with different plane information. The depth in the set of range images covers from 1 to 8 meters. The computational time of these three methods is an average among 1000 frames.

TABLE I. Processing Time

Method	Processing Time (ms)
Optimization-Based Method	241.52
RANSAC Method	319.85
RTNB Method	38.45

As shown in Table I, the processing time of the RTNB plane segmentation method is about 38.45ms per frame which is 6.3 times faster than the Optimization-Based method and 8.3 times faster than the RANSAC method. In addition, since we use integral images in normal estimation, the computational time is almost a constant that has no relevance to the size of smoothing region.

B. *Overall Accuracy*

Since the RTNB method is normal-based, the accuracy of the plane segmentation can be evaluated by the accuracy of the surface normal instead. Therefore, the RTNB method is compared to the Optimization-Based method in PCL to show their performance in normal estimation accuracy.

1) Various Depth: The testing points are sampled in a strictly flat vertical plane with various distances away from a horizantal-placed TOF sensor. The groud truth of the surface normal of each pixel should be (0, 0, -1). Fig. 6 shows the accuracy of these two method in eight depth levels (1~8m). The Optimization-Based method uses a fixed smoothing region, so we can figure out its poor performance of accuracy obviously when the depth is larger than 3m. Since we employed a dynamic determination for the size of smoothing region, the overall accuracy achieves in average of 90.12%.

2) Object Edge: Fig. 7 shows an comparion between the Optimization-Based method and the RTNB method under the condition of computng the surface normal at object edges. Normal estimation is conducted at the intersection of two orthogonal planes. The normal computed by the Optimization-Based method is influensed by both of the planes. It may lead to confusions in plane segmentation. Apparent improvement has been shown in RTNB method after considering the weight for the neighboring pixels based on step edge detectors.

V. Conclusion

In this paper we present the RTNB plane segmentation method which can be used in a navigation system for the visually impaired to avoid indoor obstacles. The RTNB method is based on fast and precise surface normal estimation. The use of integral images enhances the processing efficiency in normal estimation. Since the input range images contain noise, a dynamic determination for smoothing region and a weight assignment approach are proposed to increase the overall accuracy under the condition of a wide range of depth and object edges. Compared with the state-of-the-art Optimization-Based method and RANSAC method in PCL, the RTNB plane segmentation method is 6.3 times faster than the Optimization-Based method while 8.3 times faster than the RANSAC method. The RTNB plane segmentation method also presents an apparent improvement in accuracy in wide range of depth and object edges. We demonstrate that the RTNB method makes it possible to obtain a real-time and robust navigation system with range images from commercial TOF sensors.

VI. Acknowledgement

This research work is supported by the Important National Science and Technology Specific Project of China under Grant No. 2016ZX03001022-006, the National Natural Science Foundation of China under Grant Nos. 91438113, 61402283, 61573242, 61401501 and 61304225, the National Science and Technology Major Project under Grant No. GFZX0301010708, and the Shanghai Science and Technology Committee under Grant Nos. 16DZ1100402 and 15511105100.

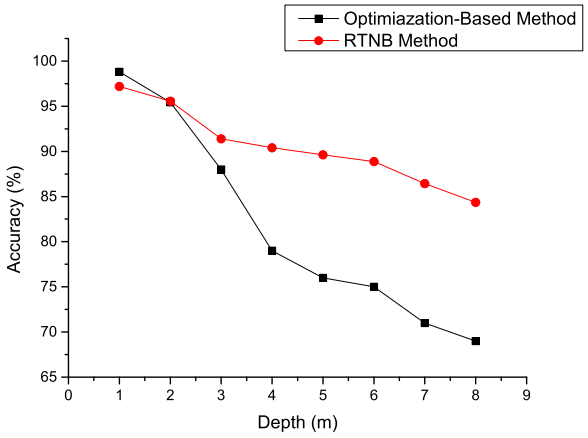
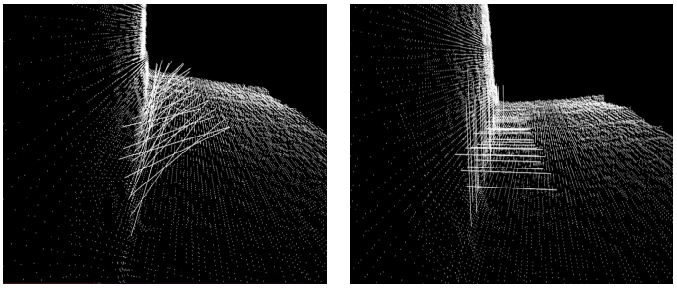


Fig. 6. The accuracy of normal estimation for various methods with respect to depth.



(a) (b)

Fig. 7. (a) Surface normal estimation at object edges using the Optimization-Based method. (b) Surface normal estimation at object edges using RTNB method.

References

[1] Holz, Dirk, et al. "Real-time plane segmentation using RGB-D cameras." Robot Soccer World Cup. Springer Berlin Heidelberg, 2011.

[2] Hulik, Rostislav, et al. "Fast and accurate plane segmentation in depth maps for indoor scenes." 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012.

[3] Gokturk, S. Burak, Hakan Yalcin, and Cyrus Bamji. "A time-of-flight depth sensor-system description, issues and solutions." Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. IEEE, 2004.

[4] Oßwald, Stefan, et al. "From 3D point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids." Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on. IEEE, 2011.

[5] Klasing, Klaas, et al. "Comparison of surface normal estimation methods for range sensing applications." Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009.

[6] Gouraud, Henri. "Continuous shading of curved surfaces." IEEE transactions on computers 100.6 (1971): 623-629.

[7] Thürrner, Grit, and Charles A. Wüthrich. "Computing vertex normals from polygonal facets." Journal of Graphics Tools 3.1 (1998): 43-46.

[8] Max, Nelson. "Weights for computing vertex normals from facet normals." Journal of Graphics Tools 4.2 (1999): 1-6.

[9] Gopi, M., Shankar Krishnan, and Cláudio T. Silva. "Surface reconstruction based on lower dimensional localized Delaunay triangulation." Computer Graphics Forum. Vol. 19. No. 3. Blackwell Publishers Ltd, 2000.

[10] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.

[11] Holzer, Stefan, et al. "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images." 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012.

[12] Lysaker, Marius, Stanley Osher, and Xue-Cheng Tai. "Noise removal using smoothed normals and surface fitting." IEEE Transactions on Image Processing 13.10 (2004): 1345-1357.

[13] Adams, Rolf, and Leanne Bischof. "Seeded region growing." IEEE Transactions on pattern analysis and machine intelligence 16.6 (1994): 641-647.