Fast, Accurate and Object Boundary-Aware Surface Normal Estimation from Depth Maps

Saed Moradi[1,3] **M.Hadi Sepanj**[1] **Alireza Memarmoghadam**[2] **Denis Laurendeau**[3] **Paul Fieguth**[1]
[1]Vision and Image Processing Group, Systems Design Engineering, University of Waterloo
[2]Department of Electrical Engineering University of Isfahan, Isfahan, Iran
[3]Computer Vision and Systems Laboratory (CVSL) Laval University Quebec, QC G1V0A6, Canada
{saed.moradi,mhsepanj,paul.fieguth}@uwaterloo.ca,denis.laurendeau@gel.ulaval.ca

Abstract

In this paper, we introduce a fast and precise surface normal estimation technique designed explicitly for depth maps, also known as organized point clouds. Our approach formulates the surface normal estimation as a closed-form expression, effectively mitigating the impact of measurement noise through multi-directional averaging. We then streamline the multi-directional normal estimation process for efficiency. Additionally, we propose a straightforward yet powerful method to eliminate inaccurate normal estimations at depth discontinuities, making our approach object boundary-aware. Comparative analyses with established surface normal estimation algorithms reveal that our method not only excels in accuracy but also exhibits the speed required for real-time applications.

1 Introduction

Surface normal vector estimation is a fundamental process in various 3D vision and 3D processing tasks, such as 3D surface reconstruction [1], scene segmentation [2], object recognition [3], and more [4–6]. It plays a crucial role in ensuring the accuracy of the complete 3D processing pipeline. As such, achieving fast and precise normal estimations is of paramount importance for practical applications.

Several approaches exist for normal estimation in the literature, which vary depending on the type of input 3D data. For unorganized point clouds, a common approach involves plane-PCA [7], where a plane is fitted to neighboring points, and the smallest eigenvalue is used to determine the surface normal vector. Although this method is accurate, its high computational complexity limits its suitability for large-scale point clouds. Consequently, various other techniques have been developed to improve accuracy and computational efficiency.

In some research, robust statistics and randomized Hough Transform are utilized to enhance normal estimation accuracy [8, 9], while others explore GPU-based implementations to accelerate the process [10]. Techniques like Deterministic MM-estimator (DetMM) are employed to reduce outliers [11]. Recently, deep learning-based methods [12–14] have gained attention, but they often require richly labeled datasets [4, 15]. In contrast to unorganized point clouds, surface normal estimation directly from depth maps (organized point clouds) has been less explored. However, this approach offers advantages such as eliminating the need to determine neighborhood points and allowing for efficient 2D image processing operators, which are notably faster than their 3D counterparts.

To address the challenge of estimating surface normals from depth maps, various methods have been proposed in the literature [16–19]. Some prior work encountered issues related to tangent vector selection and empirical parameter tuning, which led to inaccuracies, particularly for small, highly curved objects. However, other methods have achieved both accuracy and speed by employing GPU acceleration. Despite these advancements, challenges remained in estimating normal vectors for uniform areas.

In our previous work [20], a fast and accurate surface normal estimation method is proposed. In that work, a closed-form expression is proposed for each component of the surface normal vectors. Also, the method is capable of multi-scale implementation which, in turn, decreases the effect of the measurement noise. However, using a multi-scale approach increases the execution time of the algorithm. To address this issue, a fast and accurate surface normal estimation method is presented in this paper. The contributions of this work are as follows:

1. The averaging process in the multi-scale approach is used in a multi-direction manner to suppress the effect of measurement noise.
2. The multi-direction method is implemented in an efficient manner.
3. The erroneous estimated normal vectors are excluded from the final normal map using a simple yet effective method.

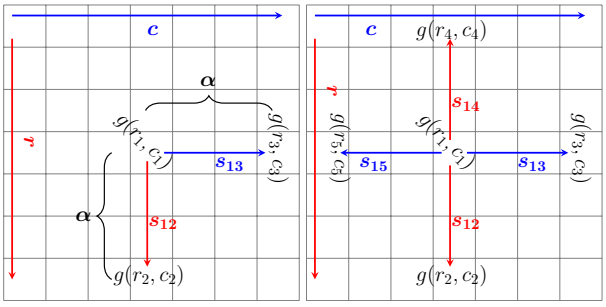
The rest of this paper is organized as follows: In Section 2, our previous work is reviewed briefly as motivation for the current work. In Section 3, the proposed method is explained in detail. Section 4 is dedicated to experiments and results. Finally, the paper is concluded in Section 5.

2 motivation and Background

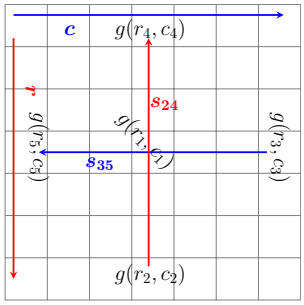
In our prior work [20], we introduced a fast method for direct surface normal vector estimation from depth maps. In this approach, we begin by constructing the projection of two surface tangent vectors within the depth map, as illustrated in Figure 1a. The surface normal vector at the query point is subsequently determined as the cross product of these two tangent vectors. The closed-form solution for these normal vectors is derived as follows:

$$\begin{aligned} n_x &= -\frac{\alpha}{f_y} d_3 (d_2 - d_1), \quad n_y = -\frac{\alpha}{f_x} d_2 (d_3 - d_1) \\ n_z &= \frac{\alpha}{f_x} v_1 d_2 (d_3 - d_1) + \frac{\alpha}{f_y} u_1 d_3 (d_2 - d_1) + \frac{\alpha^2}{f_x f_y} d_2 d_3 \end{aligned} \tag{1}$$

where, f_x and f_y denote the focal lengths. Also, $d_i = g(r_i, c_i)$, $u_i = \frac{r_i - o_x}{f_x}$, $v_i = \frac{c_i - o_y}{f_y}$. o_x and o_y are the coordinates of the optical center.



(a) 2D projection of surface tangent vectors on the depth map [20]. *(b)* Projection of surface tangent vectors construction in all four main directions.



(c) Projection of final surface tangent vectors.

Fig. 1: Construction of surface tangent vectors.

In case of noisy input, the averaging process on multi-scale results will reduce the effect of noise. Therefore (for $i = 1, 2, \cdots, k$):

$$
\begin{aligned}
n_x &= \frac{-1}{k} \sum_{i=1}^{K} \frac{\alpha_i}{f_y} d_3 \left(d_2 - d_1\right) \\
n_y &= \frac{-1}{k} \sum_{i=1}^{K} \frac{\alpha_i}{f_x} d_2 \left(d_3 - d_1\right) \\
n_z &= \frac{1}{k} \sum_{i=1}^{K} \frac{\alpha_i}{f_x} v_1 d_2 \left(d_3 - d_1\right) + \frac{\alpha_i}{f_y} u_1 d_3 \left(d_2 - d_1\right) + \frac{\alpha_i^2}{f_x f_y} d_2 d_3
\end{aligned}
\tag{2}
$$

While the single-scale version of this method is both fast and capable of accurately estimating normal vectors for smooth surfaces, the multi-scale normal estimation suffers from reduced speed, slowing down by a factor equal to the number of scales. Additionally, it fails to account for the impact of depth discontinuities at object boundaries. In the forthcoming section, we tackle both of these limitations observed in our previous work. We introduce an advanced surface normal vector estimation method that is not only fast and precise but also takes into consideration object boundaries, making it a comprehensive solution for accurate 3D surface analysis.

3 Methodology

3.1 Fast normal estimation

In our previous work [20], we employed a multi-scale approach to mitigate the influence of measurement noise on the final estimated surface normal vectors. The application of an averaging operation across different scales effectively diminishes the noise's impact. However, utilizing k scales for the final normal vector computation results in a time overhead, scaling linearly with k. To leverage the benefits of the averaging operation for noise reduction, we adopt a strategy involving multiple pairs of distinct tangent vectors to estimate the surface normal at a given query point. Subsequently, the ultimate normal vector for each point is derived by averaging the resulting normal vectors. Figure 1b illustrates the projection of four diverse surface tangent vectors on depth maps. It's worth noting that, in this context, only pairs of perpendicular tangent vectors are considered. The average of all resulting normal vectors serves as the final normal vector for each query point. This averaging process bestows robustness upon the estimation, rendering it resilient to measurement noise.

Considering all four tangent vector pairs, the normal vector can be estimated as:

$$
\begin{aligned}
\mathbf{n} &= 0.25(\mathbf{n}_{23} + \mathbf{n}_{34} + \mathbf{n}_{45} + \mathbf{n}_{52}) \\
&= 0.25(\mathbf{s}_{12} \times \mathbf{s}_{13} + \mathbf{s}_{13} \times \mathbf{s}_{14} + s_{14} \times \mathbf{s}_{15} + \mathbf{s}_{15} \times \mathbf{s}_{12}) \\
&= 0.25(\mathbf{s}_{12} \times \mathbf{s}_{13} - \mathbf{s}_{14} \times \mathbf{s}_{13} + \mathbf{s}_{14} \times \mathbf{s}_{15} - \mathbf{s}_{12} \times \mathbf{s}_{15}) \\
&= 0.25(\mathbf{s}_{12} - \mathbf{s}_{14}) \times \mathbf{s}_{13} + (\mathbf{s}_{14} - \mathbf{s}_{12}) \times \mathbf{s}_{15}) \\
&= 0.25(\mathbf{s}_{12} - \mathbf{s}_{14}) \times \mathbf{s}_{13} - (\mathbf{s}_{12} - \mathbf{s}_{14}) \times \mathbf{s}_{15}) \\
&= 0.25((\mathbf{s}_{12} - \mathbf{s}_{14}) \times (\mathbf{s}_{13} - \mathbf{s}_{15})) \\
&= 0.25(\mathbf{s}_{24} \times \mathbf{s}_{35})
\end{aligned}
\tag{3}
$$

where, $\mathbf{s}_{24}$ and $\mathbf{s}_{35}$ are two tangent vectors which are depicted in Figure 1c. Equation 3 proves that the result of the summation of four different cross products can be achieved using a single cross product. This means that using this approach can accelerate the normal estimation process by a factor of 4. Finally, the closed-form solution for the normal vectors can be determined as:

$$
\mathbf{s}_{24} = \begin{bmatrix} x_4 - x_2 \\ y_4 - y_2 \\ z_4 - z_2 \end{bmatrix} = \begin{bmatrix} u_4 d_4 - u_2 d_2 \\ v_4 d_4 - v_2 d_2 \\ d_4 - d_2 \end{bmatrix} \tag{4}
$$

$$
\mathbf{s}_{35} = \begin{bmatrix} x_5 - x_3 \\ y_5 - y_3 \\ z_5 - z_3 \end{bmatrix} = \begin{bmatrix} u_5 d_5 - u_3 d_3 \\ v_5 d_5 - v_3 d_3 \\ d_5 - d_3 \end{bmatrix} \tag{5}
$$

$$
\begin{aligned}
n_x &= -\frac{\alpha}{4 f_y} \left(d_3 + d_5\right) \left(d_2 - d_4\right), \quad n_y = -\frac{\alpha}{4 f_x} \left(d_2 + d_4\right) \left(d_3 - d_5\right) \\
n_z &= -u_1 n_x - v_1 n_y + \frac{\alpha^2}{4 f_x f_y} \left(d_2 + d_4\right) \left(d_3 + d_5\right)
\end{aligned}
\tag{6}
$$

3.2 Considering object boundaries

The proposed method, along with our prior work [20], demonstrates proficiency when addressing points associated with smooth surfaces that lack depth discontinuities. However, challenges emerge at object boundaries, where at least one of the surface tangent vectors may become invalid. As a result, the orientation of the estimated normal vector can deviate from the actual direction. To confront this issue, a novel approach is introduced herein. Surface normals are unit vectors, with their orientations representing the crucial parameters in 3D processing. Consequently, when converting a normal vector from Cartesian coordinates into spherical coordinates, only the θ component is taken into account. However, it's important to note that the length of the estimated normal vector is contingent upon the lengths of the tangent vectors, as follows:

$$
\| \mathbf{n} \|_2 \propto \| \mathbf{s}_{24} \times \mathbf{s}_{35} \|_2 \propto \| \mathbf{s}_{24} \|_2 . \| \mathbf{s}_{35} \|_2 \tag{7}
$$

As object boundaries often involve at least one tangent vector with significant length, as denoted by Equation 7, it becomes apparent that the length of the normal vector should also be substantial in such regions. Consequently, the vector's length (referred to as the r component in spherical coordinates) can serve as a valuable mask to identify and subsequently exclude erroneous normal estimations. By applying a straightforward thresholding process to the r values, outliers can be located. The final normal estimation is then carried out by applying this mask to the estimated normal map.

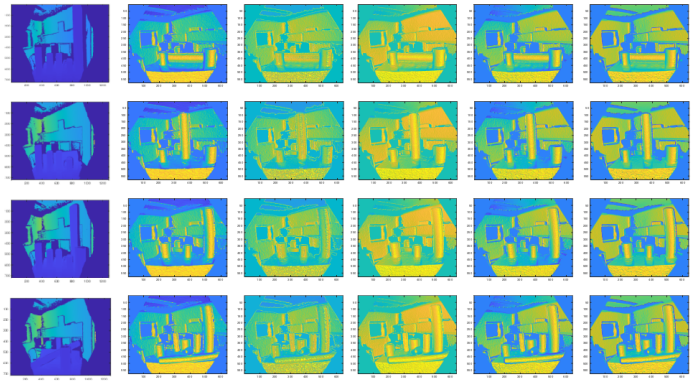
4 Experimental results

To assess the normal estimation performance of the proposed method, experiments were conducted using real data captured by a Microsoft Kinect Azure RGB-D camera, as well as synthetic depth data from the 3F2N dataset [19]. All algorithms were implemented in MATLAB. There are two approaches to visualize a surface normal vector as an image. In the first method, each component of the normal vector corresponds to a color channel, resulting in a color image derived from the depth map. Although this approach is straightforward, it may not effectively highlight minor errors in the estimated normal vectors. The second method involves converting the normal vector into spherical coordinates and examining the $\mathbf{I}_\theta$ components. The calculation of $\mathbf{I}_\theta$ is as follows:

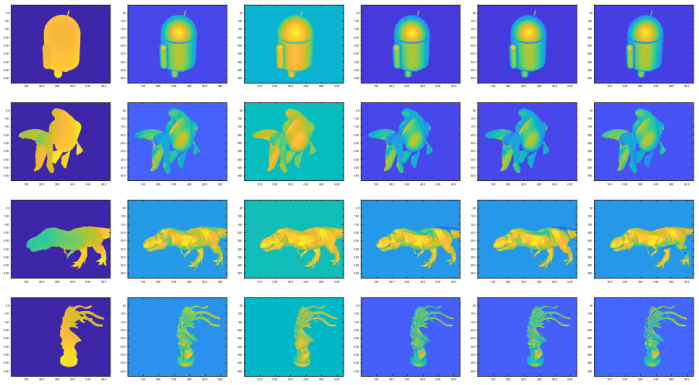
$$
\mathbf{I}_\theta = \tan^{-1} \left(\frac{\sqrt{n_x^2 + n_y^2}}{n_z} \right) \tag{8}
$$

The results of $\mathbf{I}_\theta$ for various real depth maps are illustrated in Figure 2a, revealing that the proposed method surpasses all baseline algorithms in terms of similarity to the ground truth images. Our prior work [20] secures the second position. To ensure a fair comparison, both algorithms use a common α value set to 2. Additionally, the results of the plane fitting-based method with 25 neighboring points are considered as the ground truth. Figure 2b displays the results of normal estimation on the 3F2N dataset. Once again, the proposed method and our previous work outperform other methods. Due to the synthetic images' smoothness, Nakagawa's method remains unaffected by the sensitivity of partial derivatives to noise. Consequently, this method excels on synthetic data in comparison to real data.

To perform a quantitative comparison, Mean Squared Error (MSE) is utilized as the performance metric. Table 1 and Table 2 display the MSE values for the θ image components. As observed in the tables, the proposed method exhibits superior performance when compared to all baseline algorithms, with the exception of the θ component in synthetic depth images, where no significant difference is observed among the baseline algorithms. Given the relatively smooth nature



(a) I_θ images of the estimation results of different algorithms (Real images). From left: the depth map, the normals ground truth, Fan's method [19], Nakagawa's method [18], Moradi's method [20], **the proposed method**.



(b) I_θ images of the estimation results of different algorithms (Synthetic images from 3F2N dataset [19]). From left: the depth image, the normals ground truth, Fan's method [19], Nakagawa's method [18], Moradi's method [20], **the proposed method**.

Table 1: Mean Squared error (MSE) of I_θ images

	[18]	[19]	[20]	ours
1^{st} scene	2.4227	1.8098	1.0161	**0.9301**
2^{nd} scene	2.3099	1.3769	1.0783	**0.9150**
3^{rd} scene	2.6072	1.2687	1.0244	**0.8032**
4^{th} scene	2.7241	1.2618	1.0012	**0.8918**

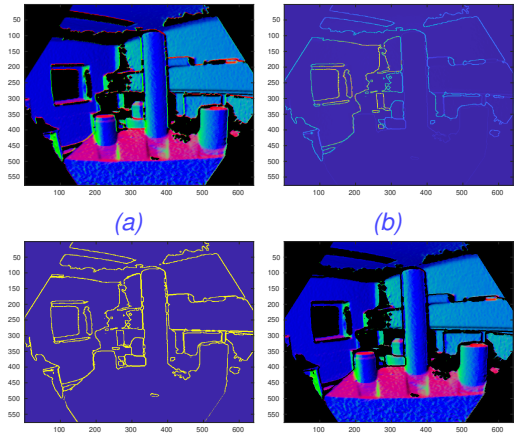
Table 2: Mean Squared error (MSE) of I_θ images (3F2N dataset [19])

	[18]	[19]	[20]	ours
1^{st} scene	0.0790	3.3966	0.0779	**0.0491**
2^{nd} scene	0.2408	2.8881	0.2378	**0.1333**
3^{rd} scene	**0.1414**	0.7437	0.1415	0.1465
4^{th} scene	0.8808	0.7672	0.8852	**0.6874**

of synthetic data, both the proposed method and our prior work [20] employ an α value set to 1. In this scenario, Nakagawa's method demonstrates similar performance to our previous work (as evident in Table 2). It's worth noting that all images are normalized within the range of $[0-2\pi]$.

The color image representation of normal vectors serves as a visual tool to highlight instances of erroneous surface normal estimation at object boundaries. This can be particularly useful for understanding the impact of depth discontinuities on the quality of surface normal estimation. In Figure 3, we present a step-by-step demonstration of the process of removing invalid normal vectors. To begin, Figure 3a displays the initial estimated surface normal vectors using Equation 3. Notably, this representation reveals areas with erroneous estimations, particularly in regions marked by depth discontinuities. As discussed in subsection 3.2, the r component (as shown in Figure 3b) in spherical coordinates tends to have large values at locations with potential erroneous estimations. Utilizing a straightforward thresholding approach on the r values, we can identify the outliers, as illustrated in Figure 3c. The final step involves applying the mask from Figure 3c to the initial normal vectors presented in Figure 3a. The result, shown in Figure 3d, represents the refined surface normal estimation, effectively excluding the erroneous estimations.

As per the table, the proposed method ranks second in terms of execution time, with our previous work (Moradi's method) [20] leading the pack. The proposed method exhibits slightly slower performance compared to our previous work. It's important to highlight that the single-scale implementation of the proposed method is equivalent to a four-scale implementation of our previous work, resulting in an almost fourfold increase in speed. The normal estimation process can be performed at an impressive rate of 135 frames per second (fps) using the proposed method. Fan's method, while straightforward in its implementation, incurs additional execution time due to the process of searching for invalid points ($\Delta Z = 0$) in the normal map.



(a) *(b)* *(c)* *(d)*

Fig. 3: Surface normal vectors refinement at object boundaries. a) initial surface normal estimation using Equation 3, b) value of r component for each point, c) outlier removal mask, and d) final result of surface normal estimation

Table 3: The average execution time for different normal estimation methods for a 576×640 depth image

Estimation method	**execution time (mS)**
local plane fitting	3776
Nakagawa's method [18]	17.132
Fan's method [19]	154.676
Moradi's method [20]	6.646
The proposed method	7.415

5 Conclusion

This paper presents an improved version of our prior work, where we optimized the normal estimation process, initially implemented efficiently as a closed-form solution. In this enhancement, we employ an averaging process for various directions to reduce the impact of measurement noise without changing the pixel distance parameter, α. We then refine the multi-directional approach for improved efficiency. This approach results in a fast and accurate normal estimation method for smooth surfaces. Nevertheless, challenges arise when it encounters object boundaries and depth discontinuities. To tackle these issues, we introduce a straightforward yet effective mask based on the length of estimated normal vectors to filter out erroneous results. Comparative analyses against established surface normal estimation algorithms, using both qualitative and quantitative assessments on real and synthetic depth images, reveal that our proposed method excels in accuracy and computational efficiency.

References

[1] D. Lu, X. Lu, Y. Sun, and J. Wang, "Deep feature-preserving normal estimation for point cloud filtering," *Computer-Aided Design*, vol. 125, p. 102860, 2020.

[2] F. Poux, C. Mattes, and L. Kobbelt, "Unsupervised segmentation of indoor 3d point cloud: application to object-based classification," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 44, no. W1-2020, pp. 111–118, 2020.

[3] H. Zhao, M. Tang, and H. Ding, "Hoppf: A novel local surface descriptor for 3d object recognition," *Pattern Recognition*, vol. 103, p. 107272, 2020.

[4] J. E. Lenssen, C. Osendorfer, and J. Masci, "Deep iterative surface normal estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 247–11 256.

[5] T. Do, K. Vuong, S. I. Roumeliotis, and H. S. Park, "Surface normal estimation of tilted images via spatial rectifier," in *European Conference on Computer Vision*. Springer, 2020, pp. 265–280.

[6] J.-W. Seo, K.-E. Kim, and K. Roh, "3d hole center and surface normal estimation in robot vision systems," in *2020 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2020, pp. 355–359.

[7] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 11, pp. 1114–1116, 1994.

[8] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang, and S. Jin, "Robust normal estimation for point clouds with sharp features," *Computers & Graphics*, vol. 34, no. 2, pp. 94–106, 2010.

[9] A. Boulch and R. Marlet, "Fast and robust normal estimation for point clouds with sharp features," in *Computer graphics forum*, vol. 31, no. 5. Wiley Online Library, 2012, pp. 1765–1774.

[10] M. Liu, F. Pomerleau, F. Colas, and R. Siegwart, "Normal estimation for pointcloud using gpu based sparse tensor voting," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2012, pp. 91–96.

[11] A. Khaloo and D. Lattanzi, "Robust normal estimation and region growing segmentation of infrastructure 3d point cloud models," *Advanced Engineering Informatics*, vol. 34, pp. 1–16, 2017.

[12] J. Zhang, J.-J. Cao, H.-R. Zhu, D.-M. Yan, and X.-P. Liu, "Geometry guided deep surface normal estimation," *Computer-Aided Design*, vol. 142, p. 103119, 2022.

[13] H. Du, X. Yan, J. Wang, D. Xie, and S. Pu, "Rethinking the approximation error in 3d surface fitting for point cloud normal estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9486–9495.

[14] S. Li, J. Zhou, B. Ma, Y.-S. Liu, and Z. Han, "Neaf: Learning neural angle fields for point normal estimation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 1, 2023, pp. 1396–1404.

[15] J. Zhou, W. Jin, M. Wang, X. Liu, Z. Li, and Z. Liu, "Improvement of normal estimation for pointclouds via simplifying surface fitting," *arXiv preprint arXiv:2104.10369*, 2021.

[16] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Asian conference on computer vision*. Springer, 2012, pp. 525–538.

[17] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab, "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2684–2689.

[18] Y. Nakagawa, H. Uchiyama, H. Nagahara, and R.-I. Taniguchi, "Estimating surface normals with depth image gradients for fast and accurate registration," in *2015 International Conference on 3D Vision*. IEEE, 2015, pp. 640–647.

[19] R. Fan, H. Wang, B. Xue, H. Huang, Y. Wang, M. Liu, and I. Pitas, "Three-filters-to-normal: An accurate and ultrafast surface normal estimator," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5405–5412, 2021.

[20] S. Moradi, D. Laurendeau, and C. Gosselin, "Multiple cylinder extraction from organized point clouds," *Sensors*, vol. 21, no. 22, p. 7630, 2021.