sensors

MDPI

Article

Multiple Cylinder Extraction from Organized Point Clouds

Saed Moradi [1,2], Denis Laurendeau [1,2,*] and Clement Gosselin [3]

[1] Department of Electrical and Computer Engineering, Faculty of Science and Engineering, Laval University, Quebec, QC G1V0A6, Canada; saed.moradi@gmail.com
[2] Computer Vision and Systems Laboratory (CVSL), Laval University, Quebec, QC G1V0A6, Canada
[3] Robotics Laboratory, Department of Mechanical Engineering, Faculty of Science and Engineering, Laval University, Quebec, QC G1V0A6, Canada; clement.gosselin@gmc.ulaval.ca
* Correspondence: Denis.Laurendeau@gel.ulaval.ca; Tel.: +1-419-656-2131 (ext. 402979)

Abstract: Most man-made objects are composed of a few basic geometric primitives (GPs) such as spheres, cylinders, planes, ellipsoids, or cones. Thus, the object recognition problem can be considered as one of geometric primitives extraction. Among the different geometric primitives, cylinders are the most frequently used GPs in real-world scenes. Therefore, cylinder detection and extraction are of great importance in 3D computer vision. Despite the rapid progress of cylinder detection algorithms, there are still two open problems in this area. First, a robust strategy is needed for the initial sample selection component of the cylinder extraction module. Second, detecting multiple cylinders simultaneously has not yet been investigated in depth. In this paper, a robust solution is provided to address these problems. The proposed solution is divided into three sub-modules. The first sub-module is a fast and accurate normal vector estimation algorithm from raw depth images. With the estimation method, a closed-form solution is provided for computing the normal vector at each point. The second sub-module benefits from the maximally stable extremal regions (MSER) feature detector to simultaneously detect cylinders present in the scene. Finally, the detected cylinders are extracted using the proposed cylinder extraction algorithm. Quantitative and qualitative results show that the proposed algorithm outperforms the baseline algorithms in each of the following areas: normal estimation, cylinder detection, and cylinder extraction.

Keywords: organized point clouds; depth map; surface normal estimation; cylinder detection; cylinder extraction

check for updates

Citation: Moradi, S.; Laurendeau, D.; Gosselin, C. Multiple Cylinder Extraction from Organized Point Clouds. *Sensors* **2021**, *21*, 7630. https://doi.org/10.3390/s21227630

Academic Editor: Stefano Berretti

Received: 11 October 2021
Accepted: 11 November 2021
Published: 17 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid development of three-dimensional (3D) scanning devices has provided a unique opportunity for robotic applications to effectively interact with the real world. Object grasping, as a common robotic task, has attracted the attention of researchers during the past decade. Most man-made objects are composed of a few geometric primitives (GPs) such as spheres, cylinders, planes, ellipsoids, or cones. Thus, object recognition can be reduced to a problem of geometric primitives extraction. Among the different geometric primitives, cylinders are the most frequently used GPs in real-word scenes [1]. Also, cylinder detection and extraction are used in several industrial applications like pipeline plant modeling [2], reverse engineering [3], automatic forest inventory [4], and 3D facility modeling [5]. This is why cylinder detection and extraction is of great importance in 3D computer vision.

There are several approaches dedicated to cylinder extraction in the literature. Hough-based methods [6–8], RANdom SAmpling Consensus (RANSAC)-based methods [1,9–11], Robust PCA [12], and quadric fitting [13] are the most popular ones for point cloud data. In [12] the authors proposed a cylinder fitting method in laser scanning point cloud data based on robust principal components analysis. After decomposition, the cylinder orientation is estimated based on the principal component corresponding to the largest

eigen-value (PC1), and PC2 as well as PC3 are used for the identification of the radius and center of the cylinder. Authors in [14] used a soft voting scheme based on curvature information to exclude outliers from cylindrical parts. In order to further remove outliers, they trained a deep-learning based classifier to filter them out. In [15], the authors projected the point cloud onto a set of directions over the unit hemisphere and tried to detect circular projections. Then the cylindrical surfaces are extracted by fitting a cylinder to each connected component. The authors in [16] used several cylinder cutting planes to obtain different ellipses. Then, RANSAC is used for both ellipse and cylinder fitting. Automated 3D pipelines recognition is investigated in [17]. The principal curvature is used as a cylinder detection algorithm and the parameters of the cylinder are extracted using RANSAC algorithm. While there is a rich literature on single cylinder extraction, multiple cylinder detection has not been investigated in depth.

In this paper, a fast and robust method is presented for multiple cylinder extraction from point cloud data. The rest of this paper is organized as follows: The research problem is defined in Section 2. In Section 3, novel solutions are proposed for the problem of multiple cylinder extraction in point cloud data. The effectiveness of the proposed solution is investigated in Section 4. Finally, the paper is concluded in Section 5.

2. Problem Definition

The problem of a cylinder extraction in point cloud data can be divided into three successive steps. At the first step, the input point cloud (3D points in Cartesian coordinates) may be represented in another feature space to deal with useful information. The orientation of surface normal vectors is one of the most common features used in 3D data processing [18–21]. There have been significant efforts dedicated to normal estimation from point cloud data in the literature. However, there is still a lack of fast and robust methods for normal estimation. This problem is investigated in depth in Section 2.1. Thereafter, the problem regarding cylinder detection and extraction is described in Section 2.2.

2.1. Normal Estimation from Point Clouds

Normal vectors estimation [22–24] is the cornerstone of many 3D computer vision tasks such as segmentation [25], registration [26], surface construction [27], object recognition [28], and others. The most common approach to estimate the surface normal vector at a point is to fit a plane to a local neighborhood of the query point and determine the vector normal to the tangent plane (See Appendix A). Numerous efforts have been made over the last decade to improve the accuracy of the surface normal estimation for unorganized point clouds. In [29], robust statistics are used to fit the optimum tangent plane for points located on high curvature surfaces. Boulch et al. [30] used Randomized Hough Transform (RHT) with statistical exploration bounds to preserve sharp features robustly. They also used a fixed-size accumulator to decrease the execution time of the estimation process. Liu et al. [31] took advantage of the results of tensor voting to decrease the estimation error. Since their algorithm has huge computational complexity, they used a GPU-based implementation in order to meet the requirements of real-time processing. In [32], the Deterministic MM-estimator (DetMM) is used to exclude outlier points for robust normal estimation. In addition to classical data processing techniques, deep learning-based methods have recently attracted the attention of the research community for surface normal vector estimation [22,33].

While many research studies dedicated to normal estimation for unorganized point clouds are reported in the literature, normal estimation directly from a depth map (organized point cloud) has received less attention. Computing normal vectors from depth images has the following advantages:

- The points in the local neighborhood of the point for which the normal is computed are known while for unorganized point clouds, an extra processing step is needed to determine the points belonging to this neighborhood.
- Most operations on organized point clouds can be performed using 2D operators and they are generally faster than 3D operations.
- The normal vectors can be computed during the scanning process.

Despite the advantages of normal estimation from depth maps, there are still some challenges. First, the input depth image is contaminated by measurement noise. Also, sharp depth discontinuities in the image can reduce the robustness of the estimated normal vectors. Fortunately, due to recent advances in scanning technology, high-quality scanners are available even at the consumer-grade level. Thus, by following a proper strategy, normal vectors can be computed directly from a raw depth map in a fast and robust manner.

Some research in the literature focuses on the estimation of normal vectors directly from input depth images. Tang et al. [34] proposed a closed-form solution for normal vectors at each point. However, erroneous formulation in tangent vectors and approximation of first-order derivatives led to a poor result in terms of accuracy. Holzer et al. [35] used an adaptive neighboring size for each point to increase the accuracy of normal estimation in sharp edges. They also used integral images for the sake of computational efficiency. The first problem related to their method is that no analytical method for the determination of the design parameters is reported and the parameters are simply computed empirically. Their method has large errors for small objects with high surface curvature. One of the most accurate normal estimation methods from a depth map was presented by Nakagawa et al. [36]. While the tangent vector construction is performed accurately, the use of single-pixel padded approximation of first-order partial derivatives and propagation of that error after a cross product of tangent vectors led to a poor and noisy result. In Section 3, we propose a fast and robust method for the estimation of normal vectors from raw depth maps that addresses all of the above issues.

2.2. Cylinder Detection and Extraction

The previously presented research works for cylinder detection and extraction suffer from two main drawbacks:

1. They are only able to detect (or extract) one cylinder at a time. The points belonging to each detected cylinder must be removed from the point cloud before starting the detection process of the next cylinder.
2. The success or failure of the detection algorithm strongly depends on the initial point (seed) selection. Without a proper strategy for robust initial point selection, the main cylinder detection module often fails.

Depth measurement errors are present in images captured by 3D sensors including the Microsoft Kinect [37]. These errors, especially outliers, are more often present near object boundaries and affect the resulting point clouds. When a local neighborhood is constructed around a query point in order to find a geometric primitive, it should not include these outliers to avoid erroneous extraction. For instance, Figure 1 shows examples of good and poor areas for initial sample selection by a cylinder extraction algorithm. The green points are good candidates for initial sample selection, while choosing the seed point among red ones may result in failure of the cylinder extraction process. Therefore, an effective cylinder detection algorithm should be able to robustly detect good initial points and reject poor ones. As shown in Figure 1, the proper candidate points for initial sample selection are located far from object boundaries.

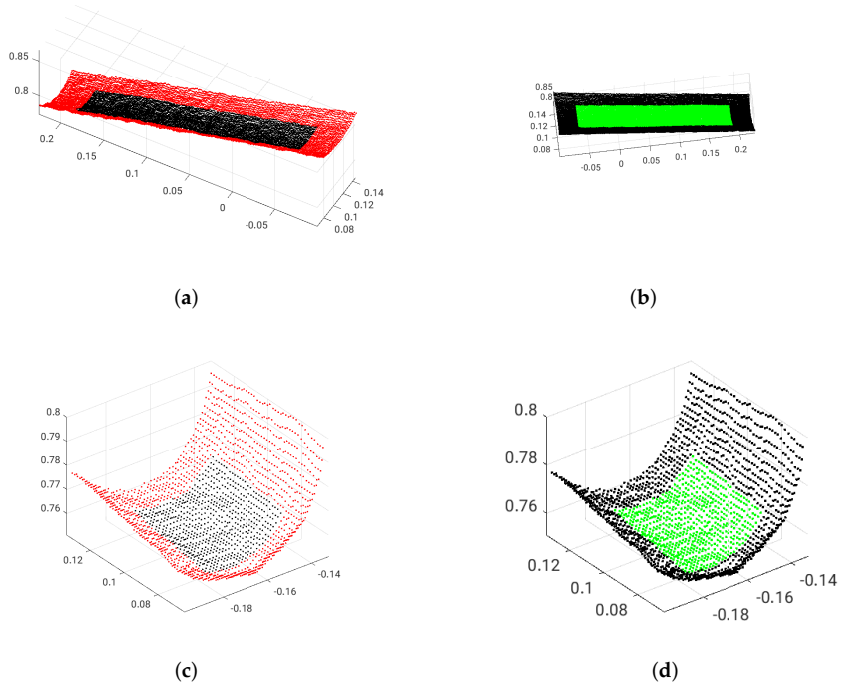


Figure 1. Examples of poor and good points for initial sample selection by a cylinder extraction algorithm. (**a**) poor initial surface points for object 1, (**b**) good initial surface points for object 1, (**c**) poor initial surface points for object 2, (**d**) good initial surface points for object 2. Red points are not good candidates for primitives extraction while green points lead to a successful extraction.

In addition to the problem relevant to the initial sample selection, multiple cylinder detection has also not been investigated in depth in the literature. Generally, a recursive process is used to detect (or extract) all of the cylinders present in the scene. To this end, after extracting the first cylinder, the points belonging to this cylinder are removed from the scene and the detection process is started over until there are no cylinders remaining in the scene. The main drawbacks of this approach are:

- The whole process must be repeated for each cylinder. The overall execution time is increased drastically when many cylinders should be extracted in the scene.
- Since the whole point cloud is modified after each detection step, the previous computations may not be reusable.
- The initial sample selection criterion should be met at each detection step.

In the next section, an approach circumventing these problems is proposed for multiple cylinder extraction.

3. Proposed Solution

In order to address the aforementioned problems, a fast and robust algorithm is presented for the detection and extraction of multiple cylinders in organized point clouds. The overall procedure of the proposed method is illustrated in Figure 2.

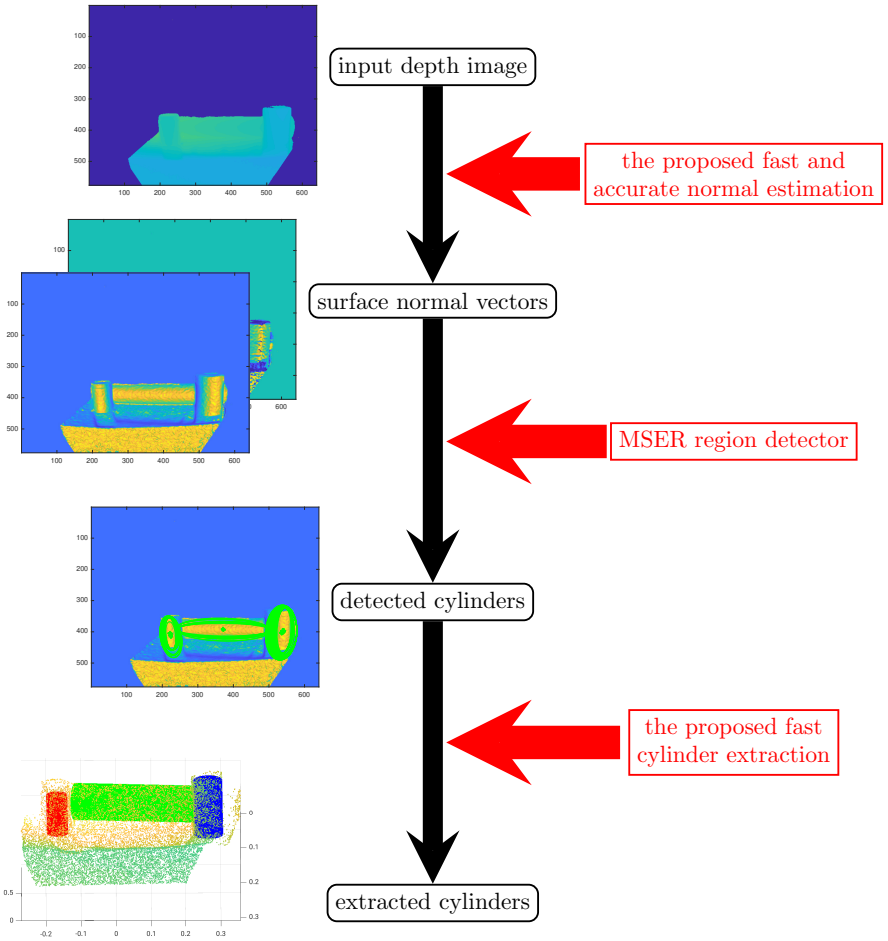


Figure 2. The overall procedure of the proposed method. The black text color indicates the data-type at each step. The required processing at each step is indicated in red.

As shown in Figure 2, in the first step, 3D surface normal vectors are estimated using a novel fast and accurate algorithm from raw input depth maps. The normal vectors are then represented in spherical coordinates to produce more distinguishable surface features. Since the normal vectors have a unit length, the third component in spherical coordinates ($\boldsymbol{I_r}$) does not contain useful information and can be discarded. Thus, the normal vectors for all points can be represented using a pair of images ($\boldsymbol{I_\phi}$ and $\boldsymbol{I_\theta}$) containing the orientation angles. Both images contain useful features to distinguish cylindrical surfaces from the rest of the scene. In the next step, the Maximally Stable Extremal Regions (MSER) feature detector is used to detect cylinders in the scene. In the final step, a fast cylinder extraction approach is proposed to estimate the parameters (axis direction and radius) of each cylinder. In the following, each of these three proposed sub-modules is explained in detail.

3.1. Fast Surface Normal Estimation

A depth image $\boldsymbol{d} = g(\boldsymbol{r}, \boldsymbol{c})$ can be converted to an organized point cloud $PC(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ using camera calibration information (Figure 3). The following equations are used for this conversion:

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{d}, \\ \boldsymbol{x} &= (\boldsymbol{r} - o_x) \odot \left(\frac{\boldsymbol{d}}{f_x}\right), \\ \boldsymbol{y} &= (\boldsymbol{c} - o_y) \odot \left(\frac{\boldsymbol{d}}{f_y}\right), \end{aligned} \tag{1}$$

where o_x and o_y are the coordinates of the principal points (the optical center), f_x and f_y denote the focal lengths, and $\odot$ stands for element-wise multiplication. Therefore, every single pixel in a depth map corresponds to a position in 3D world. The most trivial way to

estimate the surface normal vector is to compute the cross product of two perpendicular tangent vectors. Considering a smooth surface, the tangent vectors can be constructed from a depth map as shown in Figure 4. Thus, the surface normal vector is expressed as:

$$\boldsymbol{n} = \begin{bmatrix} \boldsymbol{n_x} \\ \boldsymbol{n_y} \\ \boldsymbol{n_z} \end{bmatrix} = \boldsymbol{s_{12}} \times \boldsymbol{s_{13}} \tag{2}$$

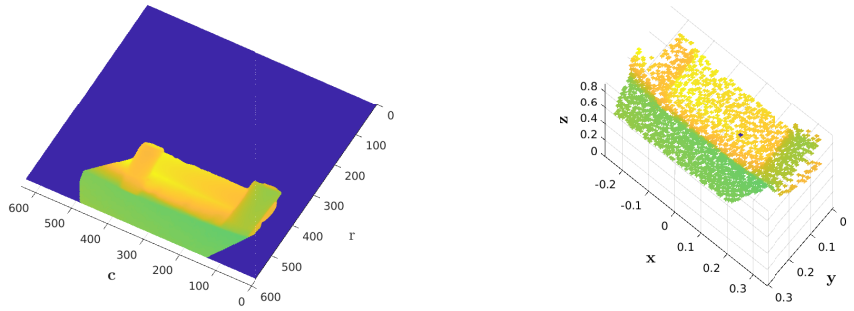
where, $\boldsymbol{s_{12}}$ and $\boldsymbol{s_{13}}$ are two tangent vectors and determined as:

$$\boldsymbol{s_{12}} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} = \begin{bmatrix} u_2 d_2 - u_1 d_1 \\ v_2 d_2 - v_1 d_1 \\ d_2 - d_1 \end{bmatrix} \quad , \quad \boldsymbol{s_{13}} = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix} = \begin{bmatrix} u_3 d_3 - u_1 d_1 \\ v_3 d_3 - v_1 d_1 \\ d_3 - d_1 \end{bmatrix} \tag{3}$$

where, $d_i = g(r_i, c_i)$, $u_i = \frac{r_i - o_x}{f_x}$, $v_i = \frac{c_i - o_y}{f_y}$. According to Figure 4, the following equations hold:

$$v_2 = v_1 \quad , \quad u_1 = u_3 \quad , \quad v_3 = v_1 + \frac{\alpha}{f_y} \quad , \quad u_2 = u_1 + \frac{\alpha}{f_x} \tag{4}$$

In Equation (4), α denotes the pixel distance between two points (Figure 4).



(**a**) (**b**)

Figure 3. Correspondence between a depth map (**a**) and an organized point cloud (**b**).

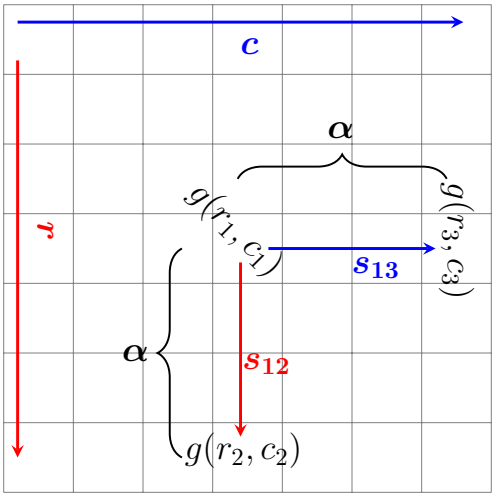


Figure 4. surface tangent vectors construction from the depth map.

Using Equation (2), each component of the normal vectors can be separately calculated as follows:

$$n_x = (v_2 d_2 - v_1 d_1)(d_3 - d_1) - (v_3 d_3 - v_1 d_1)(d_2 - d_1) \tag{5}$$

$$n_y = (u_3 d_3 - u_1 d_1)(d_2 - d_1) - (u_2 d_2 - u_1 d_1)(d_3 - d_1) \tag{6}$$

$$n_z = (u_2 d_2 - u_1 d_1)(v_3 d_3 - v_1 d_1) - (v_2 d_2 - v_1 d_1)(u_3 d_3 - u_1 d_1) \tag{7}$$

Equation (5), Equation (6) and Equation (7) can be simplified using auxiliary equations in Equation (4). The final expressions are:

$$n_x = -\frac{\alpha}{f_y} d_3 (d_2 - d_1) \tag{8}$$

$$n_y = -\frac{\alpha}{f_x} d_2 (d_3 - d_1) \tag{9}$$

$$n_z = \frac{\alpha}{f_x} v_1 d_2 (d_3 - d_1) + \frac{\alpha}{f_y} u_1 d_3 (d_2 - d_1) + \frac{\alpha^2}{f_x f_y} d_2 d_3 \tag{10}$$

In case of noisy input, the averaging process on multi-scale (using different distance values α) results will suppress the noise effect. Therefore:

$$n_x = \frac{1}{K} \sum_{i=1}^{K} n_{x_i} \quad n_{x_i} = -\frac{\alpha_i}{f_y} d_3 (d_2 - d_1) \quad i = 1, 2, \cdots, K \tag{11}$$

$$n_y = \frac{1}{K} \sum_{i=1}^{K} n_{y_i} \quad n_{y_i} = -\frac{\alpha_i}{f_x} d_2 (d_3 - d_1) \quad i = 1, 2, \cdots, K \tag{12}$$

$$n_z = \frac{1}{K} \sum_{i=1}^{K} n_{z_i} \quad n_{z_i} = \frac{\alpha_i}{f_x} v_1 d_2 (d_3 - d_1) + \frac{\alpha_i}{f_y} u_1 d_3 (d_2 - d_1) + \frac{\alpha_i^2}{f_x f_y} d_2 d_3 \quad i = 1, 2, \cdots, K \tag{13}$$

Finally, the orientation angles I_ϕ and I_θ can be calculated as:

$$I_\phi = \tan^{-1}\left(\frac{n_y}{n_x}\right) \tag{14}$$

$$I_\theta = \tan^{-1}\left(\frac{\sqrt{n_x^2 + n_y^2}}{n_z}\right) \tag{15}$$

3.2. Cylinder Detection Module

The orientation of the normal vectors is a useful extrinsic surface feature to distinguish among different types of geometric primitives. Typically, the dynamic range of the different orientation angles is split into non-overlapping bins. Then, the histogram of normals with these angles is used as the feature vector to train a classifier [18,21]. In this work, instead of the histogram of the orientation of the normal vectors, the different angles are considered as different *images*. To this end, the organized normal vectors $(\boldsymbol{n_x}, \boldsymbol{n_y}, \boldsymbol{n_z})$ in Cartesian coordinates are converted to spherical coordinates $(\boldsymbol{I_r}, \boldsymbol{I_\phi}, \boldsymbol{I_\theta})$. Since the surface normals have a unit length, the first component of the normal vector in spherical coordinates does not hold useful information concerning the surface geometry. Therefore, the two remaining components $(\boldsymbol{I_\phi}, \boldsymbol{I_\theta})$ are used for further processing. Figure 5 shows both $\boldsymbol{I_\phi}$ and $\boldsymbol{I_\theta}$ images of a sample depth image. As depicted in Figure 5c,d, both $\boldsymbol{I_\phi}$ and $\boldsymbol{I_\theta}$ images contain relevant information that can be used to distinguish between cylindrical and non-cylindrical areas. Cylindrical surfaces appear as a maximally stable elliptical region in the $\boldsymbol{I_\theta}$ image, while

there are sharp edges along the symmetry line of each cylinder in the I_ϕ image. Considering these two observations, a new cylinder detection approach is presented in the following.

The elliptical region in the I_θ image can be easily detected using the maximally stable extremal regions (MSER) feature detector due to following reasons:

- The good sample points (see Figure 1 for the definition of good sample points) of the cylindrical surfaces have a small deviation of θ (in the I_θ image). Thus, regions belonging to a cylindrical surface remain stable over a certain range of threshold values.
- Since there are small angular differences between the Z-axis and the normal vectors of good sample points, these points lie on local maxima pixels in the I_θ image.

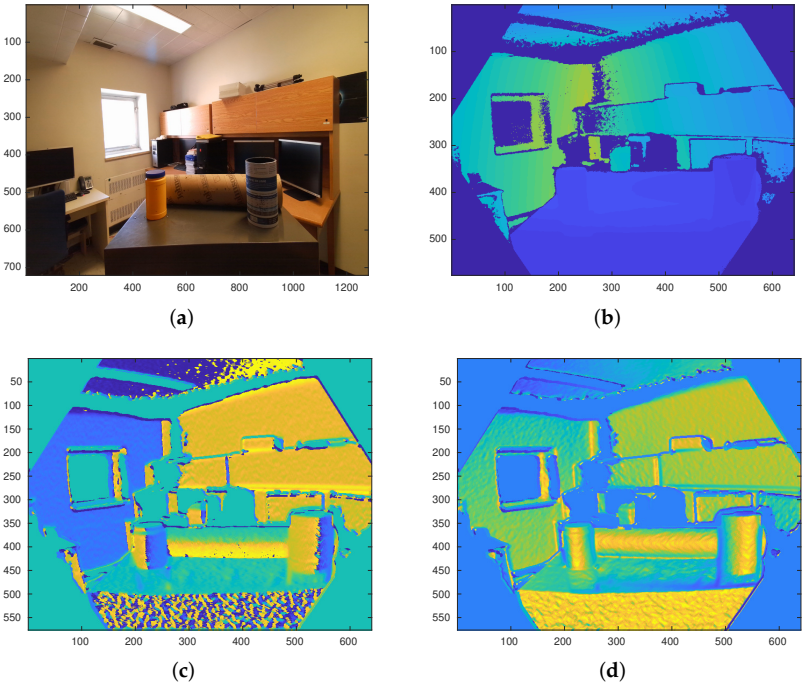


Figure 5. Different images from the same scene. (**a**) RGB, (**b**) Depth, (**c**) I_ϕ, and (**d**) I_θ images.

Therefore, these regions are both stable and extremal which can efficiently be found using MSER feature detection (In this work, a MATLAB built-in function is used to detect MSER features and obtain MSER regions. The function returns a pixel list and orientation of each region). Figure 6 shows the detected MSERs in the I_θ image. In case of a false response detection, the I_ϕ image can be used to refine the results.

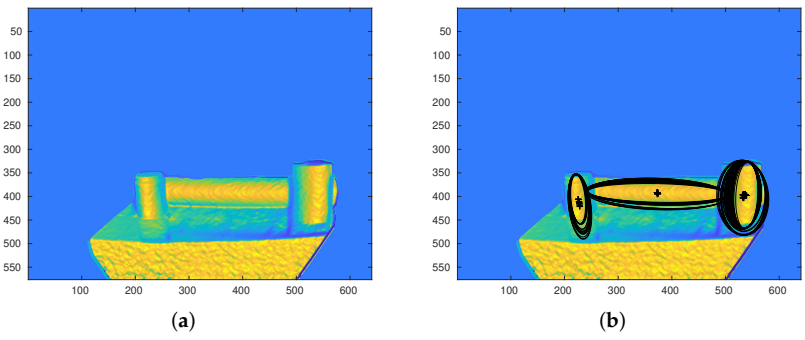


Figure 6. (**a**) I_θ image after applying a range filter, (**b**) detected MSERs.

3.3. Cylinder Extraction Sub-Module

The most straightforward method of extracting a cylinder from a set of inlier points in a point cloud is proposed by Tran et al. [1]. Their approach consists of two steps. In the first step, the orientation axis of the cylinder is found. Then, all of the inlier points are projected onto a 2D plane normal to the orientation axis. Thus, the cylinder extraction problem is relaxed into a 2D circle fitting. In Tran's method, the vector being the most orthogonal to all of the normal vectors is considered as the orientation axis of the cylinder. Identifying the orientation axis using this method requires the construction of a scattering matrix and eigenvalue decomposition which increases the execution time of the algorithm. In our method, we rather use the result of the detection step to identify the orientation axis of each cylinder. The process is described in Figure 7. As shown in the figure, the rotation angle of the MSER ellipse resulting from the detection step is used to rotate the MSER patch into a vertical patch. Then, a morphological dilation operator is used to fill holes in the MSER patch. After determining the bounding box for the region, two points on the line that passes across the bounding box are chosen as the start and the end points of the orientation axis. The reminder of the extraction process consists in determining the parameters of the projected circle in the 2D space of the plane normal to orientation axis. In this work, after a distance-based outlier removal, the Kåsa method is used for circle fitting [38].

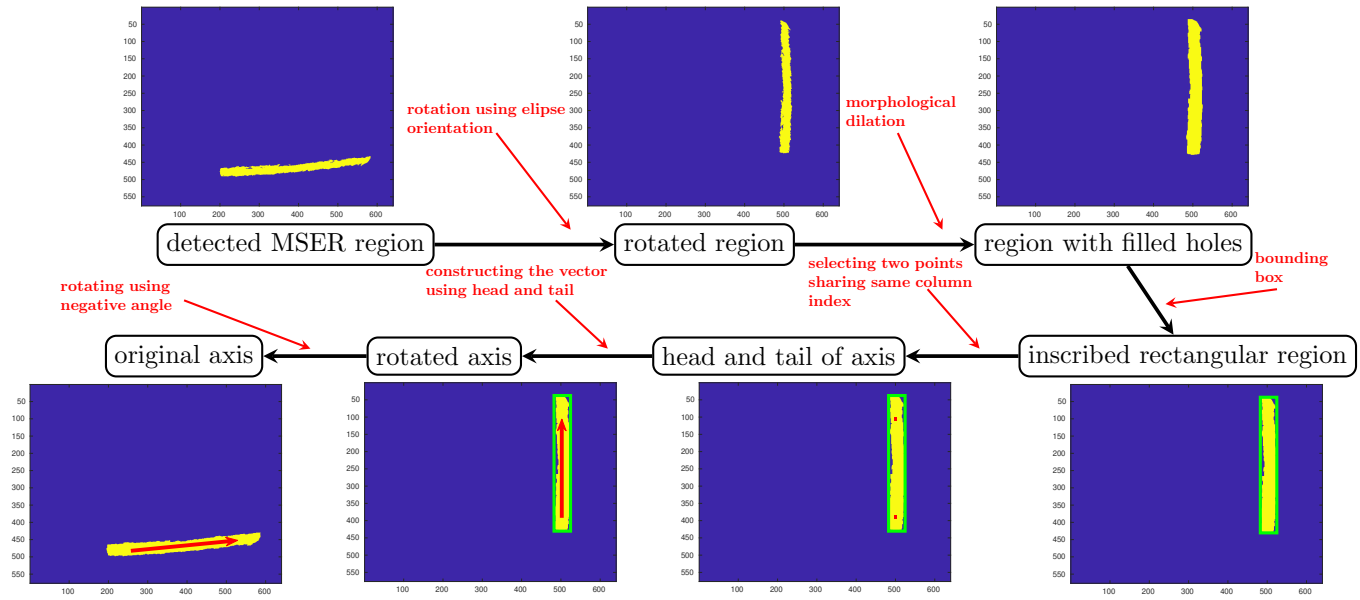


Figure 7. Estimating the orientation of the axis of a cylinder.

4. Results

In order to evaluate the cylinder extraction performance of the proposed method, some experiments were carried out on real data captured by a Microsoft Kinect Azure RGB-D camera. The comparisons and evaluations are divided into two parts. The first part (Section 4.1) is dedicated to the normal vector estimation results. Detection and extraction of the cylinders in the scene are presented in the second part (Section 4.2).

4.1. Normal Estimation

In this section, the surface normal estimation results are presented and discussed. Figures 8 and 9 show qualitative comparisons of the different methods. Since there is no ground truth data, the normal estimation results using the local plane fitting approach is used as the ground truth. As shown in the figures, Tang's method achieves the worst performance. The $\boldsymbol{I_\theta}$ image (Figure 8) is not constructed correctly using this method, and the $\boldsymbol{I_\phi}$ image is very noisy (Figure 9) compared to the other algorithms. Nakagawa's

method achieves the second best performance among all of the algorithms. Both $\boldsymbol{I_\theta}$ and $\boldsymbol{I_\phi}$ components of the normal vectors have an acceptable quality. In summary, the normal estimation method proposed in this paper outperforms other baseline algorithms. As shown in the figures, both $\boldsymbol{I_\theta}$ and $\boldsymbol{I_\phi}$ components have the most similarity to the ground truth. The quantitative results of the Mean Squared Errors (MSE) and Structural SIMilarity (SSIM) index prove the effectiveness of the proposed normal estimation method (Tables 1–4).



Figure 8. $\boldsymbol{I_\theta}$ images of the estimation results of different algorithms. From left: the ground truth, our method, Tang's method [34], Holzer's method [35], Nakagawa's method [36].

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

	Tang's Method	Holzer's Method	Nakagawa's Method	Ours
1st scene	1.1384	1.6177	1.4101	**0.1070**
2nd scene	1.1324	1.4699	1.4168	**0.1161**
3rd scene	1.1571	1.4582	1.3739	**0.1050**
4nd scene	1.3298	1.3845	1.2800	**0.1020**

Table 2. Mean Squared error (MSE) of I_ϕ images.

	Tang's Method	Holzer's Method	Nakagawa's Method	Ours
1st scene	1.8582	1.8288	4.7264	**1.6845**
2nd scene	1.9365	1.9926	4.7174	**1.7900**
3rd scene	2.0040	2.0192	4.8129	**1.8063**
4th scene	2.1487	2.1993	4.8944	**1.9430**

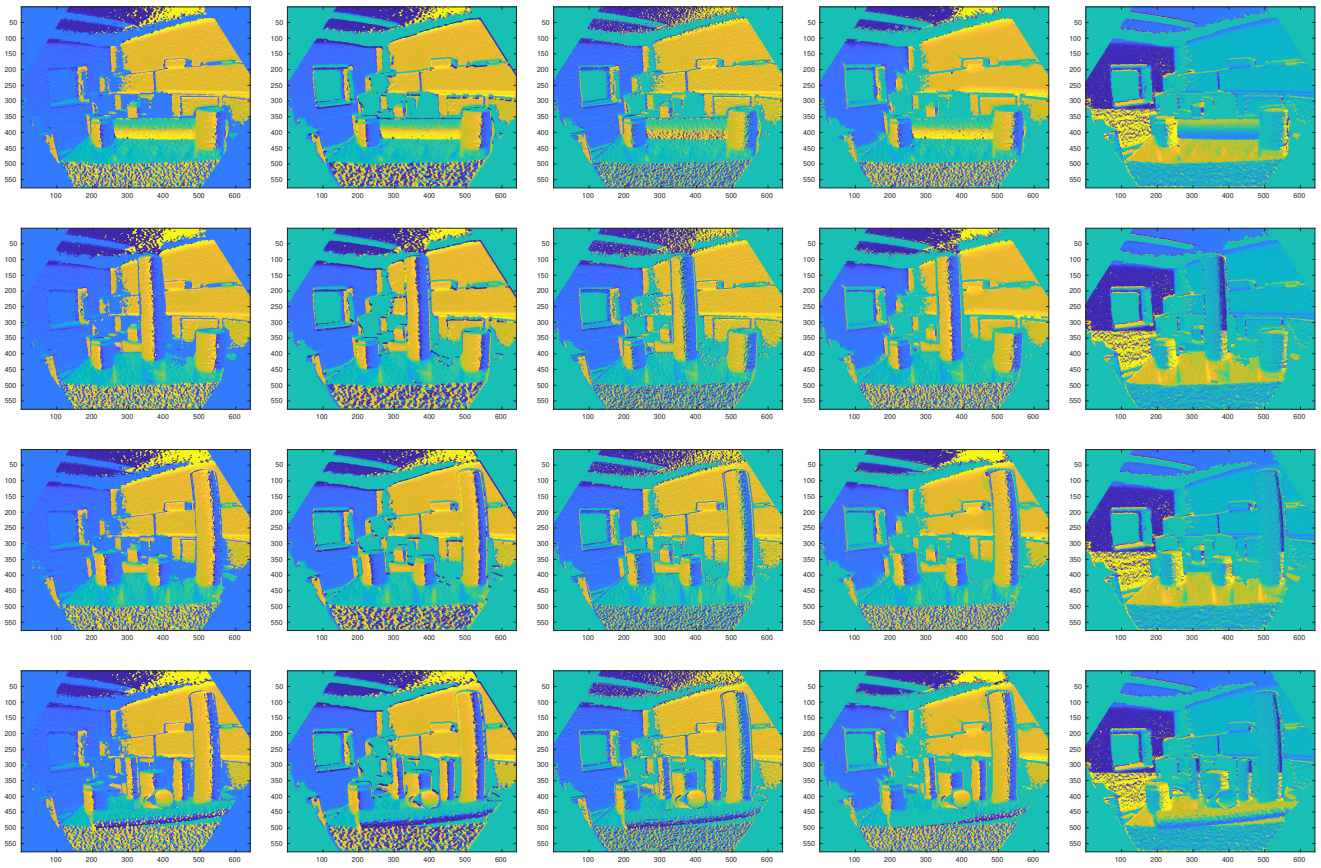


Figure 9. I_ϕ images of the estimation results of different algorithms. From left: the ground truth, our method, Tang's method [34], Holzer's method [35], Nakagawa's method [36].

Table 3. Structural Similarity Index (SSIM) of I_θ images.

	Tang's Method	Holzer's Method	Nakagawa's Method	Ours
1st scene	0.0008	0.3818	0.4066	**0.6862**
2nd scene	0.0007	0.3828	0.3984	**0.6707**
3rd scene	0.0010	0.3895	0.4092	**0.6776**
4nd scene	0.0011	0.3973	0.4241	**0.6800**

Table 4. Structural Similarity Index (SSIM) of I_ϕ images.

	Tang's Method	Holzer's Method	Nakagawa's Method	Ours
1st scene	0.5278	0.4373	0.1230	**0.6243**
2nd scene	0.5221	0.4354	0.1241	**0.6186**
3rd scene	0.5199	0.4340	0.1247	**0.6154**
4nd scene	0.4942	0.4122	0.1338	**0.5876**

In order to provide a quantitative comparison in term of computational efficiency, all of the algorithms were implemented in MATLAB. The full specifications of the simulation environment are reported in Table 5. Table 6 shows the execution time for each algorithm. As reported in the table, the proposed algorithm is the second fastest algorithm after Tang's method. Based on these results, the normal estimation process can be performed at 66 frames per second on a CPU-based implementation on a not so recent computer (See Table 5).

Table 5. The full specifications of the implementation environment.

Operating System	**MS Windows 10**
MATLAB version	2021a
Size of the test image	576 × 640
data type	double precision 64 bit floating point
CPU	Intel CORE i7-3520M @ 2.90 GHz
Memory	8GB DDR3 @ 1600 MHz

Table 6. The average execution time for different normal estimation methods for a 576 × 640 depth image.

Estimation Method	**Execution Time (mS)**
local plane fitting (considered as ground truth)	5843
Tang's method [34]	5
Holzer's method [35]	932
Nakagawa's method [36]	47
ours	**15**

4.2. Cylinder Detection and Extraction

In order to compare the results of cylinder detection by the different algorithms, the Radius-based Surface Descriptors (RSD) [39] and mean and Gaussian curvature-based method [1] are used as baseline algorithms. The cylinder detection results are depicted in Figure 10. As shown in the figure, both baseline algorithms fail in detecting the cylinders in the scene, while the proposed method detected them correctly without missing any.

Since there are no ground truth for the orientation axis of the cylinders, we used a simple approach to compare our method to Tran's method [1]. The object under test consists of two co-centered cylinder (Figure 11a). After points projection, the two circles corresponding to these cylinders must be clearly distinguishable. However, as depicted in Figure 11b Tran's method fails in this way, while our method has a better performance (Figure 11c). Both methods performed well in radius estimation. With the exception of two objects where the measurements were not accurate enough due to their material (Object 1 is made of highly reflective plastic) and dimensions (Points belonging to long objects which are comparable to vertical field of view are not measured accurately), the extraction process performed satisfactorily in other cases (Table 7).

Table 7. extracted radius of different objects (in mm).

	Object 1	**Object 2**	**Object 3**	**Object 4**	**Object 5**	**Object 6**
Tran's Method	28.82	35.44	49.04	33.11	36.42	41.29
Ours	28.53	34.98	53.46	32.88	37.64	43.98
real value	44.88	34.69	46.94	33.22	37.72	54.90

The complete cylinder extraction process from a raw depth image is demonstrated in Figure 12. The first and second columns of the figure show RGB and depth images of the input scene. As shown in the third column of the figure, all of the cylindrical surfaces are detected correctly.

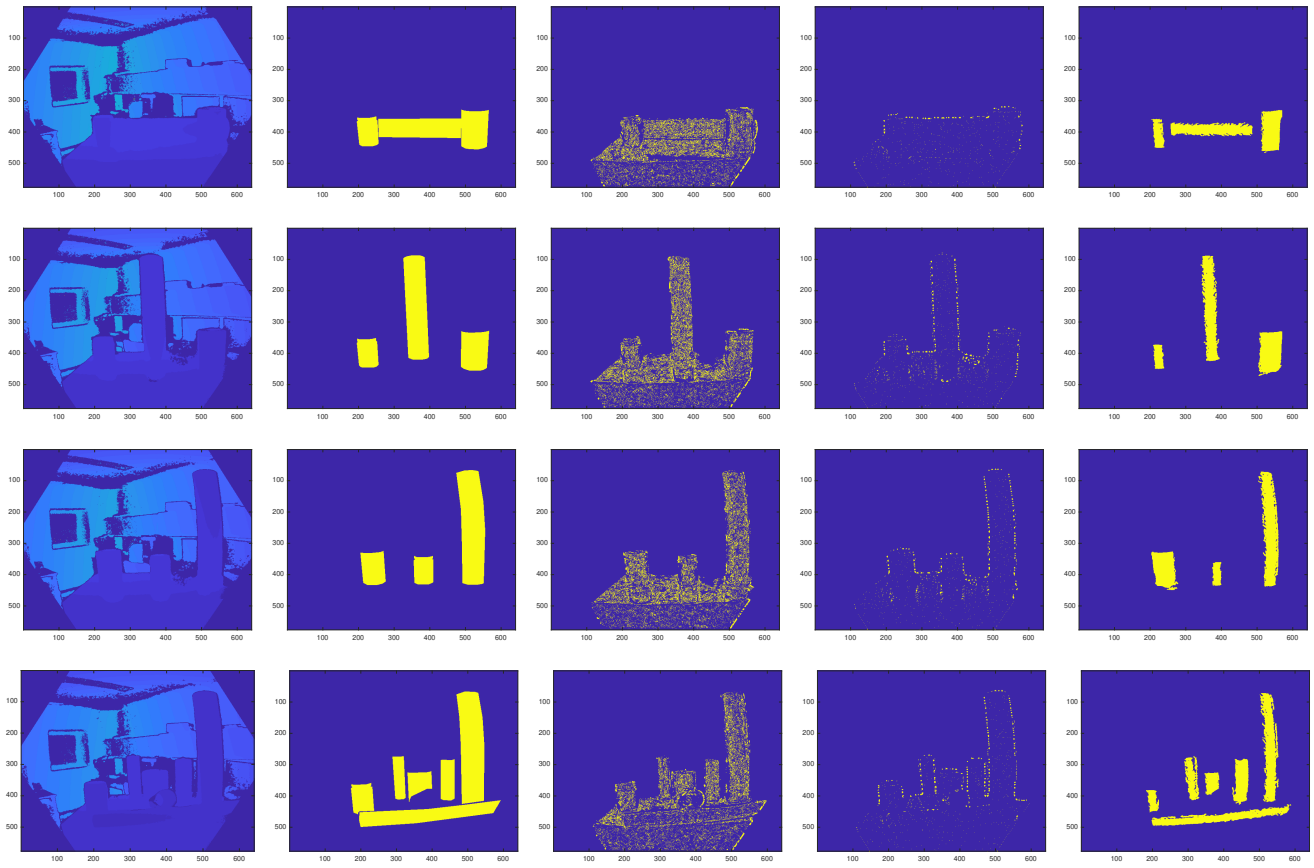
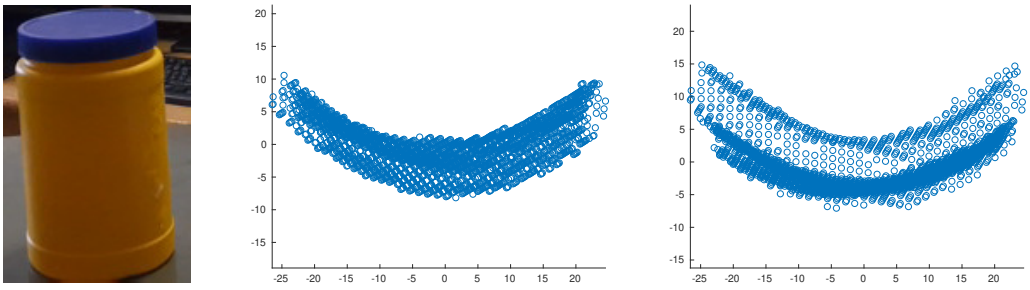


Figure 10. Cylinder segmentation results using different algorithms. From left: input depth image, ground truth, RSD-based method, principal curvature-based method, our method.



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

Figure 11. Orientation axis identification of the cylinders. (**a**) The cylindrical object, (**b**) 2D projected points resulting from Tran's method, (**c**) 2D projected points resulting from our method.

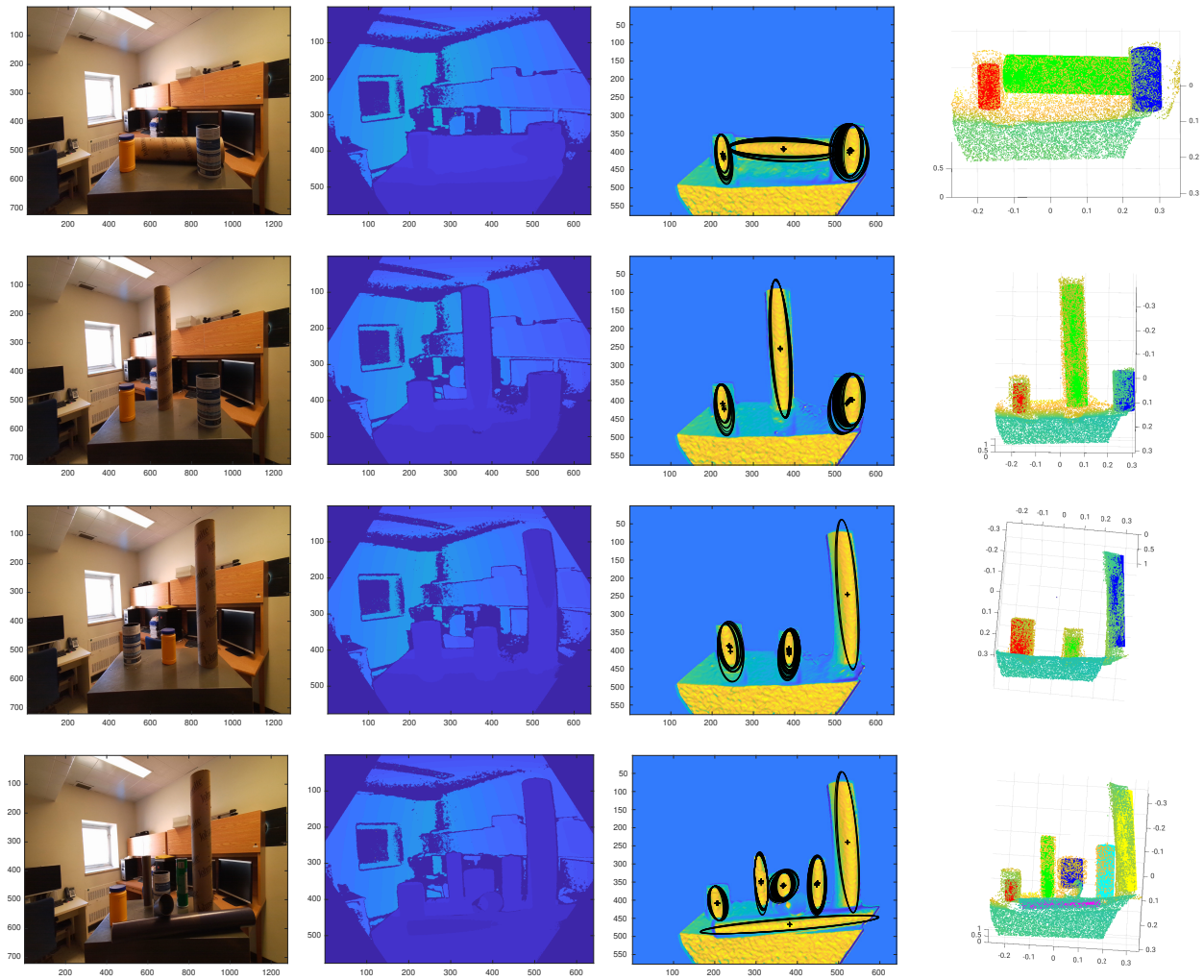


Figure 12. The first column: RGB image, the second column: depth image, the third column: detected MSERs, the fourth column: final cylinder extraction results. For our method, all cylinders are detected and extracted simultaneously.

5. Conclusions

In this paper, a new method is proposed to extract multiple cylinders from organized point clouds. Unlike the majority of the algorithms which focused on single cylinder extraction from a point cloud, all of the cylinders can be extracted from the scene using our method. Prior to the extraction step, a detection module is required to determine all of the points belonging to the different cylindrical surfaces. Since the initial sample selection plays a critical role in extracting the cylinder parameters, a straightforward approach is presented to this end, too. The overall extraction procedure consists of three sub-modules. The first module attempts to accurately estimate the normal vectors directly from the 2D depth map. Compared to the existing normal estimation methods, the proposed method can estimate the vectors in a fast and accurate manner. Considering each component of the normal vectors as an image, MSE and SSIM metrics are used to evaluate the accuracy of the proposed normal vector estimation method. Also, for a single-scale construction, the estimation method can run at 66 frame/sec rate.

After representation of the normal vectors in the spherical coordinates, a cylinder detection algorithm is proposed based on MSER feature detectors. This approach not only detects the existing cylinders in the scene simultaneously, but it also gives a pool of proper candidates (good sample points) for initial sample selection. The orientation of the query cylinder can then be easily identified using two candidate points from the pool. After identification of the orientation axis, all good sample points are projected onto a plane

along the orientation axis. Using this approach, the complex cylinder fitting problem can be solved using 2D circle fitting approaches. The extraction results on real depth maps collected during this project demonstrate the effectiveness of the proposed algorithm for real-world applications.

Author Contributions: Conceptualization, S.M. and D.L.; Formal analysis, S.M. and D.L.; Funding acquisition, C.G.; Investigation, S.M. and D.L.; Methodology, S.M. and D.L.; Project administration, D.L. and C.G.; Software, S.M.; Writing—original draft, S.M.; Writing—review and editing, S.M., D.L. and C.G. All authors have read and agreed to the published version of the manuscript.

Funding: The financial support of the Fonds de Recherche du Québec, Nature et Technologie (FRQNT) through the CHIST-ERA PeGRoGaM project is gratefully acknowledged.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank Annette Schwerdtfeger for proofreading the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Surface Normal Estimation

One of the simplest methods for surface normal estimation is based on first order plane fitting [40]. In this method, the problem of the determination of the normal reduces to the approximation of the normal to the tangent plane of the surface at a query point $\boldsymbol{p}^q$. A plane in 3D space can be represented by a point $\boldsymbol{x}$ and a normal vector $\overrightarrow{\boldsymbol{n}} = (\boldsymbol{n_x}, \boldsymbol{n_y}, \boldsymbol{n_z})$. The point to plane distance for each point $\boldsymbol{p}_i$ belonging to the neighboring set S^k (k is the number of neighboring points) can be calculated as:

$$d_i = (\boldsymbol{p}_i - \boldsymbol{x}).\overrightarrow{\boldsymbol{n}} \tag{A1}$$

$\boldsymbol{x}$ and $\overrightarrow{\boldsymbol{n}}$ can be computed using a least-squares procedure to obtain $d_i = 0$. By taking $\boldsymbol{x}$ as the centroid of the neighboring points:

$$\boldsymbol{x} = \overline{\boldsymbol{p}} = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{p}_i, \tag{A2}$$

The solution for the normal vector $\overrightarrow{\boldsymbol{n}}$ is given by computing the eigenvector corresponding to the smallest eigenvalue of the scatter matrix $\mathcal{C} \in \mathbb{R}^{3\times3}$. The scatter matrix is constructed as:

$$\mathcal{C} = \frac{1}{k}\sum_{i=1}^{k}(\boldsymbol{p}_i - \overline{\boldsymbol{p}}).(\boldsymbol{p}_i - \overline{\boldsymbol{p}})^T \tag{A3}$$

The scatter matrix $\mathcal{C}$ is symmetric and positive semi-definite and its eigenvalues are real numbers. The normal vector $\overrightarrow{\boldsymbol{n}}$ is the eigenvector of $\mathcal{C}$ corresponding to the smallest eigenvalue.

References

1. Tran, T.T.; Cao, V.T.; Laurendeau, D. Extraction of cylinders and estimation of their parameters from point clouds. *Comput. Graph.* **2015**, *46*, 345–357. [CrossRef]
2. Liu, Y.J.; Zhang, J.B.; Hou, J.C.; Ren, J.C.; Tang, W.Q. Cylinder Detection in Large-Scale Point Cloud of Pipeline Plant. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 1700–1707. [CrossRef] [PubMed]
3. Urbanic, R.; ElMaraghy, H.; ElMaraghy, W. A reverse engineering methodology for rotary components from point cloud data. *Int. J. Adv. Manuf. Technol.* **2008**, *37*, 1146–1167. [CrossRef]
4. Lalonde, J.F.; Vandapel, N.; Hebert, M. Automatic Three-Dimensional Point Cloud Processing for Forest Inventory. 2006. Available online: https://www.ri.cmu.edu/publications/automatic-three-dimensional-point-cloud-processing-for-forest-inventory/ (accessed on 11 November 2021).

5. Ahmed, M.F.; Haas, C.T.; Haas, R. Automatic detection of cylindrical objects in built facilities. *J. Comput. Civ. Eng.* **2014**, *28*, 04014009. [CrossRef]
6. Rabbani, T.; Van Den Heuvel, F. Efficient hough transform for automatic detection of cylinders in point clouds. *Isprs Wg Iii/3 Iii/4* **2005**, *3*, 60–65.
7. Tombari, F.; Di Stefano, L. Hough voting for 3d object recognition under occlusion and clutter. *IPSJ Trans. Comput. Vis. Appl.* **2012**, *4*, 20–29. [CrossRef]
8. Patil, A.K.; Holi, P.; Lee, S.K.; Chai, Y.H. An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds. *Autom. Constr.* **2017**, *75*, 65–78. [CrossRef]
9. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum* **2007**, *26*, 214–226. [CrossRef]
10. Gao, C.; Shen, Z.; Zhang, M.; Tian, Y. A RANSAC-Based Cylindrical Axis Feature Representation for Point Clouds. *J. Graph.* **2019**, *40*, 539.
11. Jin, Y.H.; Lee, W.H. Fast cylinder shape matching using random sample consensus in large scale point cloud. *Appl. Sci.* **2019**, *9*, 974. [CrossRef]
12. Nurunnabi, A.; Sadahiro, Y.; Lindenbergh, R.; Belton, D. Robust cylinder fitting in laser scanning point cloud data. *Measurement* **2019**, *138*, 632–651. [CrossRef]
13. Birdal, T.; Busam, B.; Navab, N.; Ilic, S.; Sturm, P. Generic primitive detection in point clouds using novel minimal quadric fits. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1333–1347. [CrossRef] [PubMed]
14. Figueiredo, R.; Dehban, A.; Moreno, P.; Bernardino, A.; Santos-Victor, J.; Araújo, H. A robust and efficient framework for fast cylinder detection. *Robot. Auton. Syst.* **2019**, *117*, 17–28. [CrossRef]
15. Araújo, A.M.; Oliveira, M.M. Connectivity-based cylinder detection in unorganized point clouds. *Pattern Recognit.* **2020**, *100*, 107161. [CrossRef]
16. Yu, C.; Ji, F.; Xue, J. Cutting Plane Based Cylinder Fitting Method With Incomplete Point Cloud Data for Digital Fringe Projection. *IEEE Access* **2020**, *8*, 149385–149401. [CrossRef]
17. Oh, I.; Ko, K.H. Automated recognition of 3D pipelines from point clouds. *Vis. Comput.* **2021**, *37*, 1385–1400. [CrossRef]
18. Oreifej, O.; Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 716–723.
19. Essmaeel, K.; Migniot, C.; Dipanda, A.; Gallo, L.; Damiani, E.; De Pietro, G. A new 3D descriptor for human classification: Application for human detection in a multi-kinect system. *Multimed. Tools. Appl.* **2019**, *78*, 22479–22508. [CrossRef]
20. Liu, H.; Yan, Y.; Song, K.; Chen, H. Optical challenging feature inline measurement system based on photometric stereo and HON feature extractor. *Opt. Micro- Nanometrol. VII* **2018**, *10678*, 1067812. [CrossRef]
21. Asadi-Aghbolaghi, M.; Bertiche, H.; Roig, V.; Kasaei, S.; Escalera, S. Action recognition from RGB-D data: Comparison and fusion of spatio-temporal handcrafted features and deep strategies. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 3179–3188.
22. Lenssen, J.E.; Osendorfer, C.; Masci, J. Deep iterative surface normal estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11247–11256.
23. Do, T.; Vuong, K.; Roumeliotis, S.I.; Park, H.S. *Surface Normal Estimation of Tilted Images via Spatial Rectifier*; Springer International Publishing: Cham, Switzerland, 2020; pp. 265–280.
24. Seo, J.W.; Kim, K.E.; Roh, K. 3D Hole Center and Surface Normal Estimation in Robot Vision Systems. In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020; pp. 355–359.
25. Poux, F.; Mattes, C.; Kobbelt, L. Unsupervised segmentation of indoor 3D point cloud: Application to object-based classification. *ISPRS J. Photogramm.* **2020**, *44*, 111–118. [CrossRef]
26. Quan, S.; Yang, J. Compatibility-guided sampling consensus for 3-d point cloud registration. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7380–7392. [CrossRef]
27. Lu, D.; Lu, X.; Sun, Y.; Wang, J. Deep feature-preserving normal estimation for point cloud filtering. *Comput.-Aided Des.* **2020**, *125*, 102860. [CrossRef]
28. Zhao, H.; Tang, M.; Ding, H. HoPPF: A novel local surface descriptor for 3D object recognition. *Pattern Recognit.* **2020**, *103*, 107272. [CrossRef]
29. Li, B.; Schnabel, R.; Klein, R.; Cheng, Z.; Dang, G.; Jin, S. Robust normal estimation for point clouds with sharp features. *Comput. Graph.* **2010**, *34*, 94–106. [CrossRef]
30. Boulch, A.; Marlet, R. Fast and robust normal estimation for point clouds with sharp features. *Comput. Graph. Forum* **2012**, *31*, 1765–1774. [CrossRef]
31. Liu, M.; Pomerleau, F.; Colas, F.; Siegwart, R. Normal estimation for pointcloud using GPU based sparse tensor voting. In Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012; pp. 91–96.
32. Khaloo, A.; Lattanzi, D. Robust normal estimation and region growing segmentation of infrastructure 3D point cloud models. *Adv. Eng. Inform.* **2017**, *34*, 1–16. [CrossRef]
33. Zhou, J.; Jin, W.; Wang, M.; Liu, X.; Li, Z.; Liu, Z. Improvement of Normal Estimation for PointClouds via Simplifying Surface Fitting. *arXiv* **2021**, arXiv:2104.10369.

34. Tang, S.; Wang, X.; Lv, X.; Han, T.X.; Keller, J.; He, Z.; Skubic, M.; Lao, S. *Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 525–538.
35. Holzer, S.; Rusu, R.B.; Dixon, M.; Gedikli, S.; Navab, N. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 2684–2689.
36. Nakagawa, Y.; Uchiyama, H.; Nagahara, H.; Taniguchi, R.I. Estimating surface normals with depth image gradients for fast and accurate registration. In Proceedings of the 2015 International Conference on 3D Vision, Lyon, France, 19–22 October 2015; pp. 640–647.
37. Khoshelham, K.; Elberink, S.O. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* **2012**, *12*, 1437–1454. [CrossRef] [PubMed]
38. Kåsa, I. A circle fitting procedure and its error analysis. *IEEE Trans. Instrum. Meas.* **1976**, *25*, 8–14. [CrossRef]
39. Marton, Z.C.; Pangercic, D.; Blodow, N.; Kleinehellefort, J.; Beetz, M. General 3D modelling of novel objects from a single view. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3700–3705.
40. Berkmann, J.; Caelli, T. Computation of surface geometry and segmentation using covariance techniques. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 1114–1116. [CrossRef]