

RESEARCH ARTICLE

Trajectory-Based 3D Point Cloud ROI Determination Methods for Autonomous Mobile Robot

JONG HOON PARK¹, YE EUN LIM, JUNG HYUN CHOI, AND MYUN JOONG HWANG¹, (Member, IEEE)

Department of Mechanical and Information Engineering, University of Seoul, Seoul 02504, South Korea

Corresponding author: Myun Joong Hwang (mjhwang@uos.ac.kr)

This work was supported by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) through Smart Farm Innovation Technology Development Program funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) under Grant 421031-04.

ABSTRACT With depth cameras and LiDAR improving and generating more data, their applications in 3D point clouds are growing rapidly. However, the vast amount of generated data increases the computational load and results in a shortage of storage space. Therefore, a preprocessing step to reduce the number of points is required before using the 3D point cloud. This study proposes region of interest (ROI) determination methods that sequentially construct circular and rectangular ROIs along the target trajectory of the robot to extract only crucial data for the target task. These two ROI determination methods have two benefits. First, they maintain the resolution of the raw data; second, they create two ROIs that match perfectly regardless of the complexity of the trajectory. To verify the high performance of these two ROI determination methods, we conducted simulations and experiments using various data; artificial frames, keyframes, and sequential frames. As a result, when the distance between the center points was small, 25% of the diameter or height of the circular and rectangular ROIs, the classification evaluation results were closer to 1 and the processing speed was faster than the raw data acquisition rate. However, we confirm that there is a trade-off relationship between the classification results and the processing time according to the distance parameter. In addition, through the qualitative comparison with the previous study, the long cuboid ROI determination method, we identified the limitations of the previous study and the advantages of the two proposed ROI determination methods.

INDEX TERMS 3D point cloud, preprocessing for data reduction, circular and rectangular ROIs, trajectory.

I. INTRODUCTION

A point cloud comprises a set of points based on a specific coordinate system. In the three-dimensional (3D) point cloud [1], each point has Euclidean coordinates for the three axes X, Y, and Z. Therefore, it is possible to represent the surfaces of various objects in 3D space.

3D point cloud data can be generated using various sensors. First, they can be acquired through LiDAR (Light Detection And Ranging), which measures the distance to surrounding

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu¹.

objects based on the time reflected by a light source. One study [2] obtained omnidirectional 3D point cloud data in an intersection environment, and another study [3] obtained data corresponding to 90° forward in a virtual environment. In addition, 3D point cloud data can be acquired simultaneously with RGB images using depth cameras. Depth cameras are largely divided into three types based on the method of measuring depth information: the stereotype estimates the depth by forming a parallax with two images; the time-of-flight (ToF) type measures the depth through transmission and reception of optical signals, and the structural light type calculates the depth by projecting structural light. Existing

studies have acquired 3D point cloud data using stereo [4], ToF [5], and structural-light-type [6] depth cameras. Furthermore, more sophisticated 3D point cloud data can be obtained by fusing heterogeneous sensors. A previous study used fused data from a 2D LiDAR and a depth camera [7].

The use of 3D point clouds has grown recently, as sensor accuracy and resolution have increased and their costs have decreased. Many studies in robotics and autonomous driving are using them for various tasks. Some studies use only 3D point cloud data for object detection in indoor [8] and road environments [9], [10]. There are some studies that use data from LiDAR and image information to effectively sample road environments [11] and detect objects in autonomous driving conditions [12]. In another study, drones utilized point cloud data from their surroundings to generate obstacle avoidance paths for autonomous flight [13]. Study [14] implemented simultaneous localization and mapping (SLAM) technology using global point cloud maps in dynamic environments.

However, the vast amount of data increases computational load and causes storage issues when using 3D point clouds in various applications. Since thousands to tens of thousands of points are measured per frame from LiDAR or depth cameras, real-time operations become difficult when all the measured data are used on a large scale. Therefore, to efficiently utilize the 3D point cloud without these issues, extracting only the points required for the target task is necessary to reduce the load. In most cases, preprocessing for data reduction is essential before using a 3D point cloud.

There are two main methods to reduce 3D point cloud data. The first method is down-sampling to reduce the overall number of points by representing 3D points in the voxel space at regular intervals [15]. The second method involves determining a region of interest (ROI) within the 3D point cloud space and extracting only the points within the ROI. These two methods are being actively studied using various methodologies, such as the method to determine the amount of data to be extracted and the criteria to be used in determining the importance of the data.

Examining the previous studies on these methods, in the case of the down-sampling method, there is a study [16] that reduces the number of points and calculation cost by performing down-sampling through voxels when recognizing people with various postures. In addition, a previous study [17] modified this method to express and utilize 3D point clouds in various ways based on voxels. However, if the points are expressed in voxels, two problems arise. First, the resolution of the resulting data is lower than that of the original data, and the sophisticated features of the original data are compromised [18]. Second, unimportant points outside the target trajectory remain because this method reduces the overall number of points rather than extracting a portion of the original data. In the ROI determination method, if the 3D point cloud is measured from several LiDAR mounted on the vehicle to recognize the neighborhood of the moving vehicle, the ROI is determined to 200 m from the center of

the sensors to extract only the points around the vehicle [19]. Similarly, in an autonomous driving study [20] using a 3D point cloud, the number of points was reduced by determining the long cuboid centered on the vehicle as the ROI. However, determining only the area around the robot or vehicle as the ROI can result in a loss of data around the target trajectory, which can slow the response of the task. A complicated curved trajectory can lead to issues involving data that are unimportant to the target task.

This study proposes ROI determination methods for 3D point clouds to determine circular and rectangular ROIs based on the target trajectory of a robot or vehicle. The methods used in this study set the center points of the ROIs along the trajectory and extract the points within the total ROI by sequentially determining circular and rectangular ROIs based on the center points. There are two benefits of using these methods. First, extracting only the important data near the target trajectory is possible while maintaining a high resolution of the original data. Second, even if a robot or vehicle has a complex trajectory, the ROI can be determined that precisely matches that trajectory, which can quicken the target task's response.

The remainder of this paper is organized as follows. Chapter 2 explains the method for transforming the 3D point cloud into a global frame and reducing it by determining the circular and rectangular ROIs. In Chapter 3, we generate the artificial data and use it to identify the influence of the distance parameter between the center points. Also, we build the factory map and the simple trajectory in the Gazebo simulator, and verify the classification evaluation results and processing time of the proposed ROI determination methods with the keyframes obtained using a mobile robot equipped with LiDAR. In Chapter 4, we conduct the experiments by generating the complex trajectory in a real-world environment and adding GPS and IMU to the existing robot. We first validate the results on the keyframes, such as simulations, and further analyze the processing time of the two ROIs using the sequential frames. Finally, we confirm the advantages of the proposed methods by performing a qualitative comparison with the previous study, the long cuboid ROI determination method.

II. METHODOLOGY

The method to transform the points into a global frame, which is the reference coordinate system of the trajectory, is discussed first. Next, the methods that determine two ROIs, circular and rectangular, along the trajectory to reduce the 3D point cloud are explained.

A. TRANSFORMATION OF 3D POINT CLOUD

To determine whether the 3D points belong to the trajectory-based ROI, it is necessary to match the reference frame of the 3D point cloud to that of the waypoints in the trajectory. However, points measured from sensors have coordinates based on sensor frames, and waypoints typically have coordinates based on global frames in the workspace. Therefore, the

3D point cloud must be represented on a global frame through coordinate transformation.

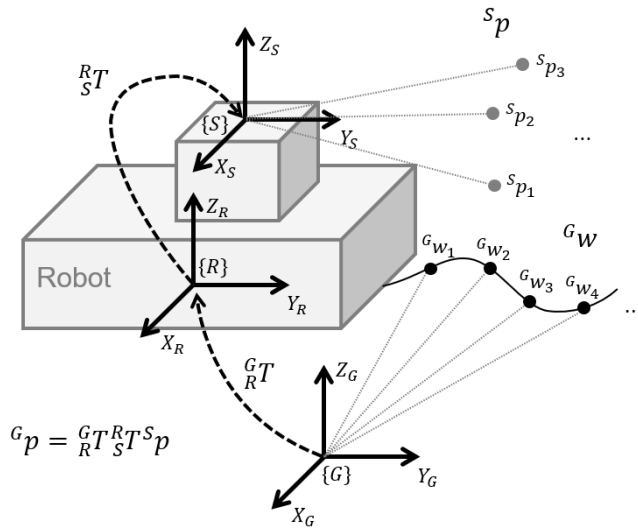


FIGURE 1. Example of transforming 3D point cloud to the global frame.

Fig. 1 shows an example of a 3D point cloud coordinate transformation, which consists of the global frame G , robot frame R , sensor frame S , 3D point cloud based on the sensor frame $S_p = \{s_{p_i} | i = 1 \text{ to } n\}$, and waypoints based on the global frame $G_w = \{g_{w_k} | k = 1 \text{ to } r\}$. To transform the 3D point cloud S_p into a global frame, the transformation matrices G_R^T and R_S^T should be used. The R_S^T matrix can be obtained through the calibration between the robot and the sensor, and the G_R^T matrix can be obtained through the localization of the robot. Thus, the 3D point cloud based on global frame G_p can be derived from the products of the two matrices.

Finally, the positional relation between the 3D point cloud based on the global frame G_p and waypoints based on the same frame G_w was identified to extract only the points within the circular and rectangular ROIs.

B. CIRCULAR ROI

The first ROI determination method proposed in this study determines a series of circular ROIs. As shown in Fig. 2, a circular ROI with a radius r is determined by selecting the center points of the ROI $c = \{c_j | j = 1 \text{ to } m\}$ among waypoints G_w in the trajectory. Then, as shown in (1), we calculate the distance $d_{X_G Y_G}$ in the $X_G Y_G$ plane between the points based on the global frame G_{p_i} and the center point of the ROI c_j , and identify whether the point G_{p_i} exists within the circular ROI by comparing the distance $d_{X_G Y_G}$ with the radius of the ROI r . If necessary, constraints can be added to the Z_G axis.

$$d_{X_G Y_G} = \left\| G_{p_i} - c_j \right\|_{X_G Y_G}$$

$$G_{p_i} = \begin{cases} in, & d_{X_G Y_G} \leq r \\ out, & d_{X_G Y_G} > r \end{cases} \quad (1)$$

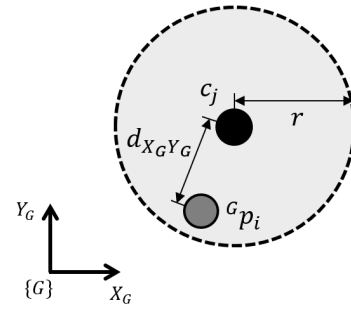


FIGURE 2. The circular ROI.

We performed the corresponding task sequentially for all points and center points of the ROI using double loops, as shown in Algorithm 1, and finally completed trajectory-based 3D point cloud reduction for the entire circular ROI, as shown in Fig. 3. If the point belongs to a specific circular ROI, the identification of the remaining circular ROI is omitted through the break, and processing is performed quickly. At this time, we can set the range of the circular ROI by adjusting the radius r , and we can adjust the amount of overlap of each circular ROI by adjusting the distance between the center points of that ROI.

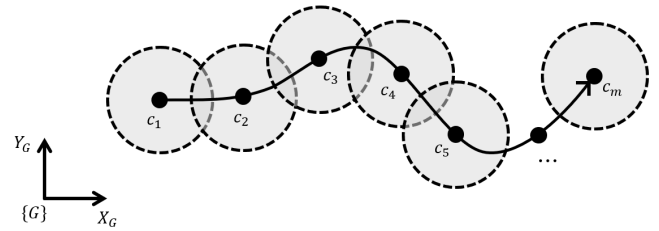


FIGURE 3. Example of the circular ROI based on the trajectory.

Algorithm 1 The Pseudo-Code for the Circular ROI

Input: 3D point cloud based on the sensor frame s_p ,
Homogeneous transformation matrices G_R^T and R_S^T ,
Set of ROI center points c ,
Radius of circular ROI r

Output: Set of the points within the ROI ROI

- 1: $n = \text{size of } S_p$
- 2: $m = \text{size of } c$
- 3: $ROI = \emptyset$
- 4: $G_p = G_R^T R_S^T S_p$
- 5: **for** $i \rightarrow 1$ to n **do**
- 6: **for** $j \rightarrow 1$ to m **do**
- 7: $d_{X_G Y_G} = \|G_{p_i} - c_j\|_{X_G Y_G}$
- 8: **if** $d_{X_G Y_G} \leq r$ **then**
- 9: $ROI.push(G_{p_i})$
- 10: **break**
- 11: **end if**
- 12: **end for**
- 13: **end for**

C. RECTANGULAR ROI

The second ROI determination method proposed in this study determines a series of rectangular ROIs. As shown on the left side of Fig. 4, a rectangular ROI $A_jB_jC_jD_j$ with height a , width b , and tangent angle $\theta = \{\theta_j | j = 1 \text{ to } m\}$ is determined by selecting the center points of ROI c among the waypoints ${}^G w$ in the trajectory. Using (2), we calculate the vector $\vec{u} = c_j \vec{m}$ with tangent angle θ_j and vector $\vec{v} = m \vec{C}_j$ perpendicular to vector \vec{u} . Subsequently, using our calculated figures, we obtain vertices A_j, B_j and C_j , and vectors $\vec{A}_j \vec{B}_j, \vec{A}_j \vec{C}_j, \vec{B}_j \vec{C}_j$ and $B_j \vec{C}_j p_i$. At this point, vector \vec{v} may correspond to the opposite vector $m \vec{D}_j$; however, in that case, vertices B_j, C_j and A_j, D_j change and eventually produce the same result. Finally, as shown on the right side of Fig. 4, we extract only the points within the rectangular ROI through the inner products of the vectors. As with the circular ROI, constraints on the Z_G axis can be added if required.

$$\begin{aligned} \vec{u} &= \frac{a}{2} (\cos \theta_j, \sin \theta_j) = (u_x, u_y) \\ \vec{v} &= \frac{b}{2 \|\vec{u}\|} (u_y, -u_x) \\ A_j &= (c_{j,x}, c_{j,y}) - \vec{u} - \vec{v} \\ B_j &= (c_{j,x}, c_{j,y}) - \vec{u} + \vec{v} \\ C_j &= (c_{j,x}, c_{j,y}) + \vec{u} + \vec{v} \\ G p_i &= \begin{cases} \text{in,} & 0 \leq \vec{A}_j \vec{B}_j \cdot \vec{A}_j \vec{G} p_i \leq \|\vec{A}_j \vec{B}_j\|^2 \\ & \text{and } 0 \leq \vec{B}_j \vec{C}_j \cdot \vec{B}_j \vec{G} p_i \leq \|\vec{B}_j \vec{C}_j\|^2 \\ \text{out,} & \text{otherwise} \end{cases} \quad (2) \end{aligned}$$

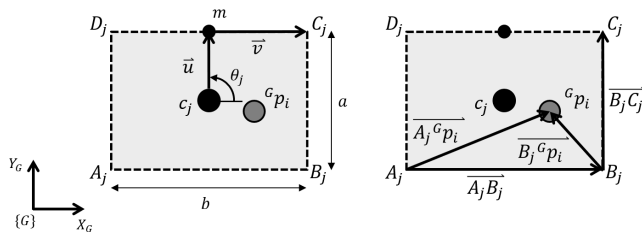


FIGURE 4. The rectangular ROI.

Similar to the circular ROI, we repeated the corresponding task through the double loops of Algorithm 2 and completed the reduction for the rectangular ROI, as shown in Fig. 5. We also performed quick processing using a break. For the rectangular ROI, we can set the range of the ROI by adjusting the height a and the width b ; the amount of overlap of each rectangular ROI can be adjusted by changing the distance between the center points.

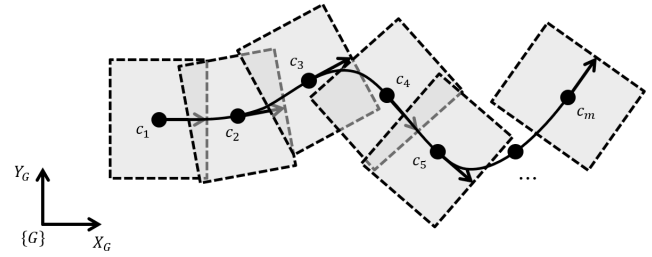


FIGURE 5. Example of the rectangular ROI based on the trajectory.

Algorithm 2 The Pseudo-Code for the Rectangular ROI

Input: 3D point cloud based on the sensor frame ${}^S p$,
Homogeneous transformation matrices ${}^G_R T$ and ${}^S_R T$,
Set of ROI center points c ,
Set of tangent angles θ ,
Height and width of rectangular ROI a and b

Output: Set of the points within the ROI ROI

- 1: $n = \text{size of } {}^S p$
- 2: $m = \text{size of } c$
- 3: $\text{ROI} = \emptyset$
- 4: ${}^G p = {}^G_R T {}^S_R T {}^S p$
- 5: **for** $j \rightarrow 1$ to m **do**
- 6: $\vec{u} = \frac{a}{2} (\cos \theta_j, \sin \theta_j) = (u_x, u_y)$
- 7: $\vec{v} = \frac{b}{2 \|\vec{u}\|} (u_y, -u_x)$
- 8: $A_j = (c_{j,x}, c_{j,y}) - \vec{u} - \vec{v}$
- 9: $B_j = (c_{j,x}, c_{j,y}) - \vec{u} + \vec{v}$
- 10: $C_j = (c_{j,x}, c_{j,y}) + \vec{u} + \vec{v}$
- 11: **end for**
- 12: **for** $i \rightarrow 1$ to n **do**
- 13: **for** $j \rightarrow 1$ to m **do**
- 14: **if** $0 \leq \vec{A}_j \vec{B}_j \cdot \vec{A}_j \vec{G} p_i \leq \|\vec{A}_j \vec{B}_j\|^2$
and $0 \leq \vec{B}_j \vec{C}_j \cdot \vec{B}_j \vec{G} p_i \leq \|\vec{B}_j \vec{C}_j\|^2$ **then**
- 15: $\text{ROI}.\text{push}({}^G p_i)$
- 16: **break**
- 17: **end if**
- 18: **end for**
- 19: **end for**

III. SIMULATION

We applied the circular and rectangular ROI determination methods proposed in this paper through simulations. First, we generated artificial point cloud data, straight and curved trajectories, to focus on the influence of the distance parameter between the center points. We applied the two ROI determination methods. Second, we constructed the factory map in the Gazebo simulator and validated the results on several keyframes.

A. SIMULATION OF THE ARTIFICIAL FRAMES

There are parameters of the diameter, height, and width in the circular and rectangular ROIs. However, since these

TABLE 1. The artificial point cloud data in the simulation.

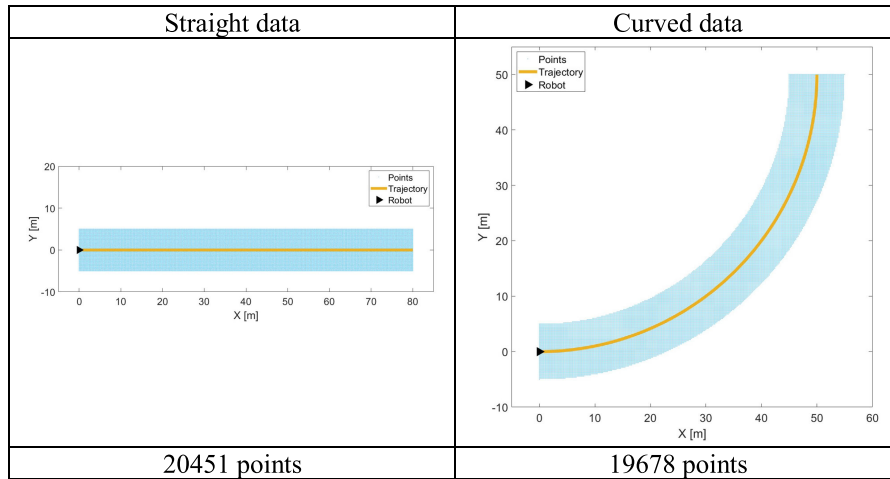


TABLE 2. The circular ROI classification results using the simulation's artificial frame data.

	The distance between the center points [m] (Ratio to the diameter of the circle [%])			
	9m (150%)	6m (100%)	3m (50%)	0.6m (10%)
Straight data				
	6234 points (30.48%)	9460 points (46.26%)	11467 points (56.07%)	11897 points (58.17%)
Curved data				
	6060 points (30.8%)	9288 points (47.2%)	11287 points (57.36%)	11777 points (59.85%)

parameters can be set in various ways as needed by the user, we fixed them at specific values and conducted the validation by focusing on the distance between the center points. At this time, we set the distance at various ratios of the diameter or height for each circular and rectangular ROIs, considering both sparse and dense ROI cases; 150%, 125%, 100%, 75%, 50%, 25%, and 10%. Therefore, we set the diameter and width to the same value to extract the same region through the two ROI determination methods and the height to half the width, which is an arbitrary value; a diameter of 6 m for the circular ROI, the height of 3 m, and width of 6 m for the rectangular ROI.

To verify the influence of the distance parameter between center points more accurately, we generated and utilized dense artificial point cloud data with uniform intervals instead of data from specific environments with multiple densities. Since circular and rectangular ROIs show different characteristics in straight and curved trajectories, we set up the

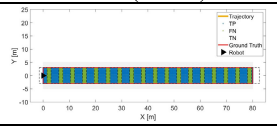
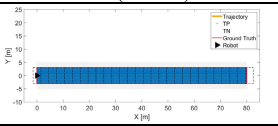
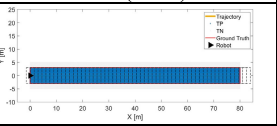
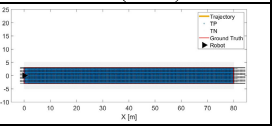
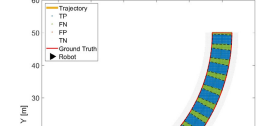
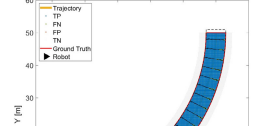
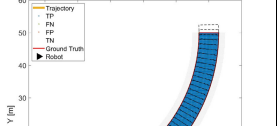
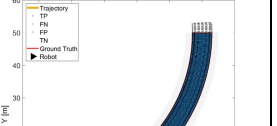
straight and curved trajectories and point cloud data as shown in Table 1. At this time, since there is no constraint on the Z-axis in this validation, for simplicity, we created the point cloud only for the XY plane by making the value of the Z-axis zero; straight data with 10 m width and 80 m length, curved data with 10 m width and 50 m radius. We also set the robot's pose to have (0, 0) position and 0 rad of heading angle.

We conducted a simulation in an Ubuntu 18.04 environment using the Intel i7-10750H CPU, and extracted points with the proposed ROI determination methods by receiving the point cloud data previously generated at 20 Hz using ROS (Robot Operating System) melodic. Also, we used classification evaluation metrics as criteria for performance verification. We first determined the ground truth ROI represented by the solid red lines in Tables 2 and 4, and labeled the point cloud data true or false to assess the accuracy of the two ROIs, proposed in this paper, determined along the trajectory while

TABLE 3. The circular ROI classification evaluation metrics and processing time using the simulation’s artificial frame data.

		The num. of ROI [EA]	TP [EA]	FP [EA]	FN [EA]	TN [EA]	Precision	Recall	Accuracy	Min/Max time [ms]	Average time [ms]
Straight data	150 %	10	6234	0	6197	8020	1	0.5015	0.6970	3.42 / 14.79	12.491
	125 %	12	7422	0	5009	8020	1	0.5971	0.7551	4.43 / 15.46	13.670
	100 %	14	9460	0	2971	8020	1	0.7610	0.8547	4.14 / 17.08	14.645
	75 %	19	10696	0	1735	8020	1	0.8604	0.9152	8.48 / 17.92	17.078
	50 %	28	11467	0	964	8020	1	0.9225	0.9529	5.34 / 20.23	18.015
	25 %	56	11683	0	748	8020	1	0.9398	0.9634	12.35 / 25.17	20.276
	10 %	139	11897	0	534	8020	1	0.9570	0.9739	11.53 / 35.88	21.563
Curved data	150 %	10	6060	0	5744	7874	1	0.5134	0.7081	3.38 / 13.66	12.471
	125 %	11	7435	0	4369	7874	1	0.6299	0.7780	3.46 / 18.84	12.850
	100 %	14	9288	0	2516	7874	1	0.7869	0.8721	4.49 / 15.82	15.152
	75 %	19	10560	0	1244	7874	1	0.8946	0.9368	5.64 / 17.6	16.480
	50 %	28	11287	0	517	7874	1	0.9562	0.9737	8.74 / 18.35	17.709
	25 %	55	11671	0	133	7874	1	0.9887	0.9932	8.57 / 24.52	19.785
	10 %	136	11777	0	27	7874	1	0.9977	0.9986	10.82 / 34.38	21.253

TABLE 4. The rectangular ROI classification results using the simulation’s artificial frame data.

	The distance between the center points [m] (Ratio to the height of the rectangle [%])			
	4.5m (150%)	3m (100%)	1.5m (50%)	0.3m (10%)
Straight data				
	8525 points (41.69%)	12431 points (60.78%)	12431 points (60.78%)	12431 points (60.78%)
Curved data				
	7900 points (40.15%)	11644 points (59.17%)	11834 points (60.14%)	11851 points (60.22%)

maintaining the previously set constant width. As a result, for all cases of circular and rectangular ROIs, we classified all points into four categories: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The classification results of the points are presented in Tables 2 and 4. We attached only four cases; 150%(Sparse case), 100%(Fit case), 50%(Dense case), 10%(Very dense case). The number of points extracted through these ROIs and the ratio of the total number are listed as well. Using (3), we calculated the precision, recall, and accuracy values, which are classification evaluation metrics [24]. In addition, to analyze the processing time, we derived the minimum, maximum, and average values using 100 repeated measurements. The number of ROIs, classification results of the points, classification evaluation metrics, and processing time are listed in

Tables 3 and 5. The number of ROIs and average processing time are presented as graphs in Figs. 6 and 7.

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN} \\
 Accuracy &= \frac{TP + TN}{TP + FP + FN + TN} \tag{3}
 \end{aligned}$$

First, looking at the circular ROI results, there is little difference between the straight and curved trajectories. As shown in Table 2, since the circular ROIs always exist inside the ground truth ROI in both trajectories, the number of FP points is 0, and the precision has a value of 1 for all cases in Table 3. In addition, as the distance between the center points

TABLE 5. The rectangular ROI classification evaluation metrics and processing time using the simulation’s artificial frame data.

		The num. of ROI [EA]	TP [EA]	FP [EA]	FN [EA]	TN [EA]	Precision	Recall	Accuracy	Min/Max time [ms]	Average time [ms]
Straight data	150 %	19	8525	0	3906	8020	1	0.6858	0.8090	5.33 / 29.54	15.676
	125 %	23	10137	0	2294	8020	1	0.8155	0.8878	5.45 / 24.37	16.194
	100 %	28	12431	0	0	8020	1	1	1	5.92 / 23.4	18.743
	75 %	37	12431	0	0	8020	1	1	1	7.5 / 26.76	20.652
	50 %	56	12431	0	0	8020	1	1	1	9.04 / 32.06	21.178
	25 %	111	12431	0	0	8020	1	1	1	13.23 / 40.64	22.758
	10 %	277	12431	0	0	8020	1	1	1	27.52 / 51.34	30.696
Curved data	150 %	19	7883	17	3921	7857	0.9978	0.6678	0.7999	4.62 / 19.17	15.697
	125 %	22	9434	26	2370	7848	0.9973	0.7992	0.8782	5.89 / 22.82	18.117
	100 %	28	11621	23	183	7851	0.9980	0.9845	0.9895	5.63 / 25.24	19.315
	75 %	37	11791	29	13	7845	0.9975	0.9989	0.9979	6.8 / 27.01	20.289
	50 %	55	11799	35	5	7839	0.9970	0.9996	0.9980	8.02 / 31.07	21.501
	25 %	109	11799	42	5	7832	0.9965	0.9996	0.9976	14.07 / 41.63	22.904
	10 %	272	11802	49	2	7825	0.9959	0.9998	0.9974	25.66 / 54.68	29.948

TABLE 6. The information of the keyframes used in the simulation.

	Position X [m]	Position Y [m]	Heading θ [rad]
Keyframe 1	18	-29.99	-3.14
Keyframe 2	-2.07	-30	-3.14
Keyframe 3	-28.02	-22.35	1.63

decreases and the circular ROIs fill more area of ground truth ROI in Table 2, the number of FN points decreases, and recall and accuracy increase in Table 3. In the case of the average processing time, there is no significant difference between the two trajectories, as shown in Table 3, since the number of points in two artificial data is similar to about 20,000. However, in Fig. 6, as the distance decreases and the number of ROIs required increases, the average processing time increases. When an excessively large number of ROIs is used like the distance value of 10%, the average time is maximized.

In the case of the rectangular ROI, unlike the results of the circular ROI, there are differences for the two trajectories. In the straight trajectory, the rectangular ROI exactly matches the ground truth ROI like the circular ROI as shown in Table 4, so the number of FP points becomes 0, and the precision has a value of 1 for all cases in Table 5. However, when the distance is less than 100% in Table 4, unlike the circular ROI, the rectangular ROI becomes the same as the ground truth ROI. So all three classification evaluation metrics have values of 1 in Table 5. Conversely, in the curved trajectory, the vertices of the rectangular ROI are outside the ground truth ROI in Table 4, so FP points always exist and the values of the precision are not exactly 1 in Table 5. In addition, the inside of the ground truth ROI in Table 4, especially the outer part of the curve, is not completely filled, so the recall

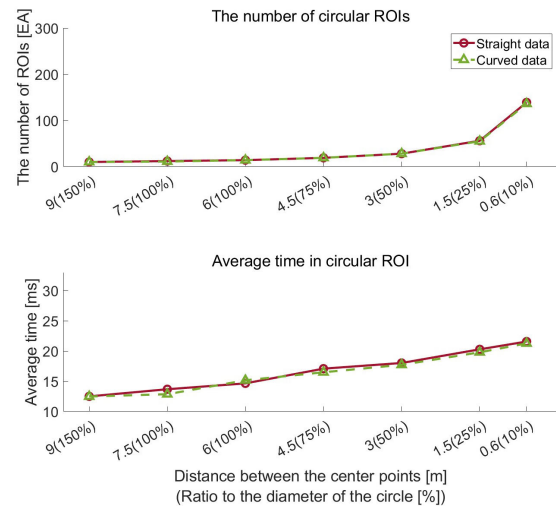


FIGURE 6. The circular ROI processing time using the simulation’s artificial frame data.

and accuracy are not exactly 1 in Table 5. But, as the distance becomes shorter and the interval between the ROIs become tighter, these values increase. In the average processing time in Table 5, the values at the two trajectories are similar to each other like circular ROI, but as the distance decreases and the number of ROI increases, as shown in Fig. 7, the average time is larger.

Through validation using the artificial data, we confirm that the distance parameter has two significant influences on the ROI results. First, when the ROI is sparsely placed by setting the distance larger than the diameter or height for the circular and rectangular ROIs, respectively, the ROI always fails to fill the ground truth ROI, resulting in low performance. In Tables 3 and 5, based on the distance of 100%, it can be seen that the classification evaluation results

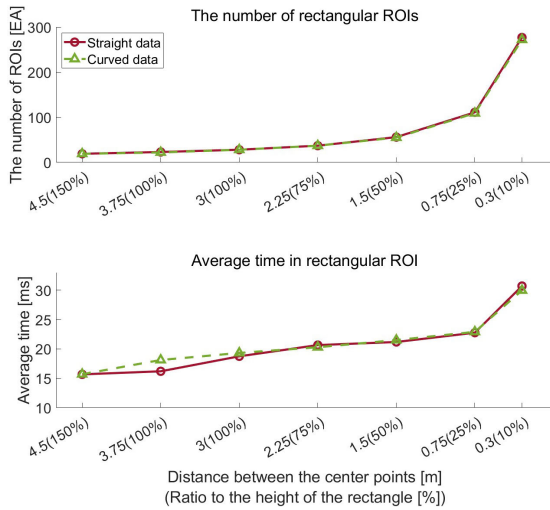


FIGURE 7. The rectangular ROI processing time using the simulation’s artificial frame data.

at the higher rate of distance are significantly lower than the lower rate of distance. Second, the distance parameter causes a trade-off relationship between the classification evaluation results and the average processing time. To avoid the problem mentioned in the first influence, if the ROI is placed very dense by unconditionally lowering the distance as shown in 10% of Tables 2 and 4, the average processing time will be very long, as shown in Figs. 6 and 7. Therefore, considering the number of points, other parameters of the ROI, and the performance of the PC, the user should set the appropriate distance value to ensure acceptable time while satisfying the required classification evaluation results. In the simulations and experiments introduced in this paper, we will use the four distance values, 100%, 75%, 50%, and 25%, excluding sparse cases where the distance is greater than 100%.

B. SIMULATION ON THE KEYFRAMES

For the simulation, we constructed a 100 m × 100 m factory map in the Gazebo simulator, as shown in Fig. 8. A simple trajectory was set, using straight lines and a curved section, such as the solid yellow line in Fig. 8. We used the differential drive type Scout 2.0 mobile robot (Fig. 9), and Velodyne’s VLP-16 LiDAR (Fig. 10). We used the ROS melodic to obtain the poses of the robot from the Gazebo simulator at 100 Hz and control the robot at a constant velocity of 0.5 m/s along the given trajectory. In addition, we obtained 3D point cloud data on the factory map using LiDAR at 20 Hz. The simulation was performed in the Ubuntu 18.04 environment using an Intel i7-10750H CPU, the same as previous artificial data validation.

To validate the performance of the two ROIs, we set up three keyframes with two straight lines and one curve as the target trajectory (Fig. 8 and Table 6). The raw 3D point cloud data projected onto the XY plane and the number of points in each keyframe are listed in Table 7. In addition, we set the parameters of the two ROIs as follows: a diameter of 28 m



FIGURE 8. The factory map of the Gazebo simulator [21].



FIGURE 9. AGILE-X’s Scout 2.0 mobile robot [22].



FIGURE 10. Velodyne’s VLP-16 LiDAR [23].

for the circular ROI, a height of 12 m, and a width of 28 m for the rectangular ROI. The center points have a minimum set length of 42 m for both ROIs. In addition, when setting the center points of the two ROIs, as mentioned earlier, the distances between them were set differently in four cases: 25%, 50%, 75%, and 100% of the diameter of the circular ROI and the height of the rectangular ROI. Therefore, as listed in Tables 8–11, we applied circular and rectangular ROIs to extract the points in the cases of three keyframes and the four distances between center points.

We used the classification evaluation metrics, the criteria used in the previous validation. Also, as in the previous method, we determined the ground truth ROI, like the solid red lines in Tables 8 and 10, and proceeded with labeling for the three keyframes. At this time, to compare only the middle

TABLE 7. The factory point cloud data of the keyframes in the simulation.

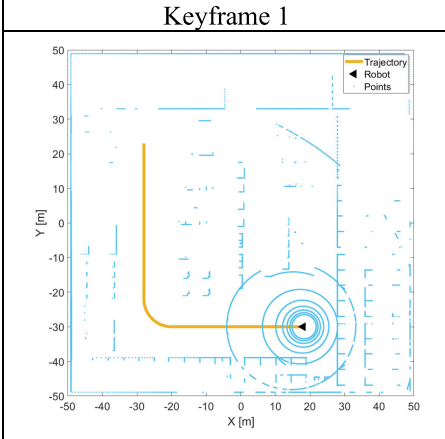
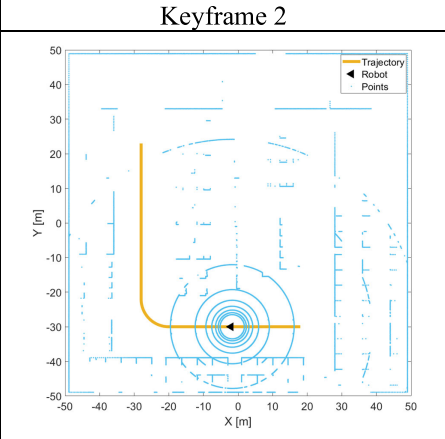
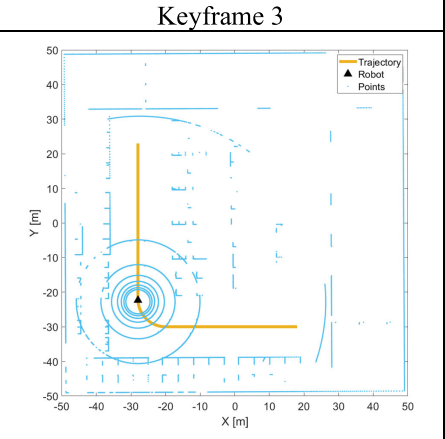
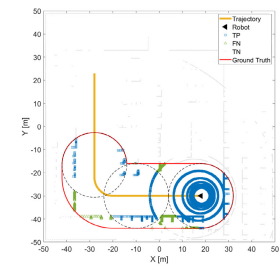
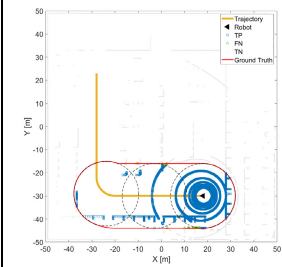
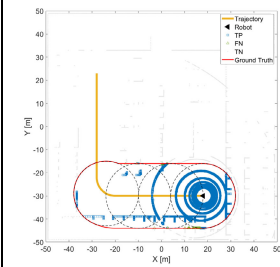
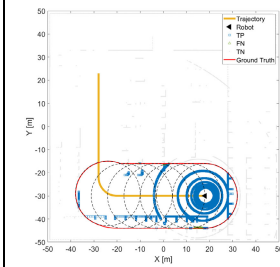
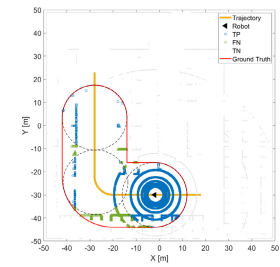
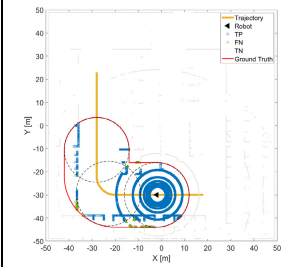
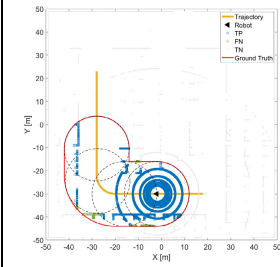
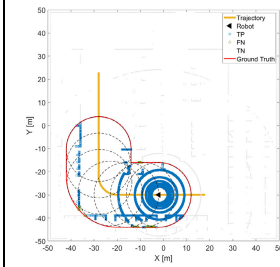
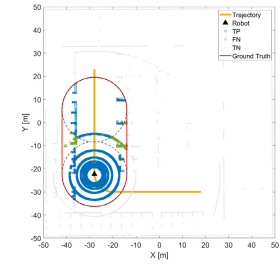
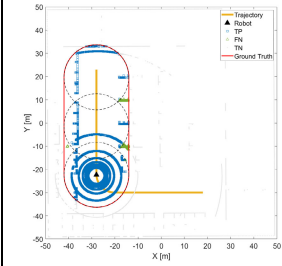
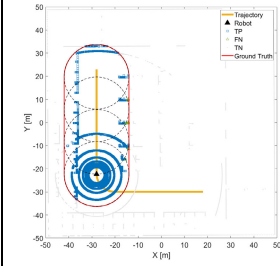
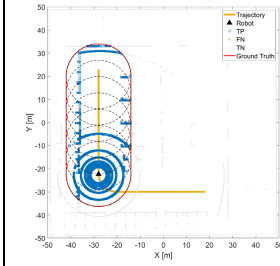
Keyframe 1	Keyframe 2	Keyframe 3
		
29087 points	29376 points	29119 points

TABLE 8. The circular ROI classification results using the simulation's keyframe data.

	The distance between the center points [m] (Ratio to the diameter of the circle [%])			
	28m (100%)	21m (75%)	14m (50%)	7m (25%)
Keyframe 1	 17370 points (59.72%)	 18233 points (62.68%)	 18248 points (62.74%)	 18271 points (62.82%)
Keyframe 2	 15813 points (53.83%)	 16882 points (57.47%)	 16907 points (57.55%)	 17051 points (58.04%)
Keyframe 3	 14875 points (51.08%)	 16027 points (55.04%)	 16386 points (56.27%)	 16476 points (56.58%)

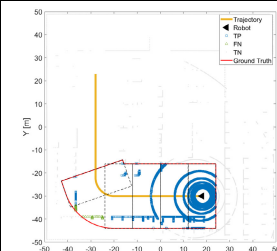
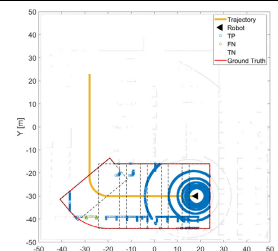
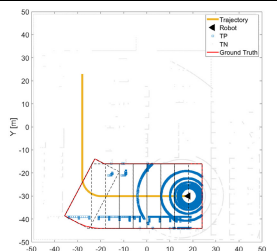
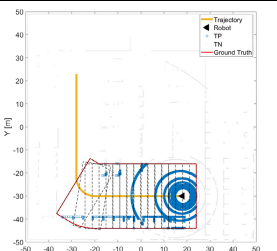
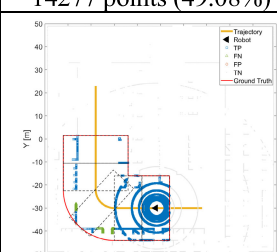
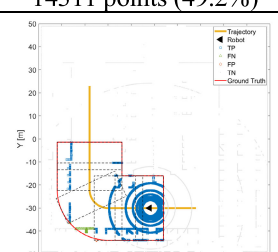
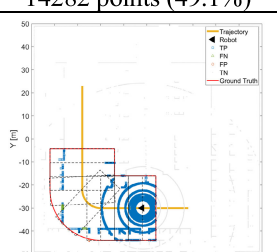
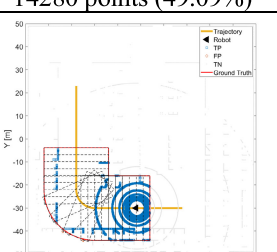
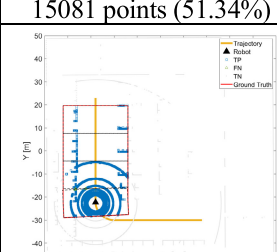
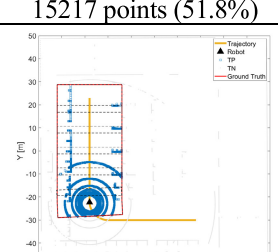
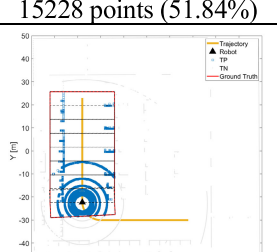
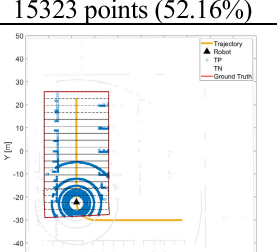
parts of the two ROIs, we matched the beginning and end parts of the ground truth ROI with the circular and rectangular ROIs, respectively. Therefore, we categorized the points into

four categories for all cases, and the results and the number of points extracted through the two ROIs are summarized in Tables 8 and 10, respectively. Finally, we calculated the evalu-

TABLE 9. The circular ROI classification evaluation metrics and processing time using the simulation’s keyframe data.

		The num. of ROI [EA]	TP [EA]	FP [EA]	FN [EA]	TN [EA]	Precision	Recall	Accuracy	Min/Max time [ms]	Average time [ms]
Keyframe 1	100 %	3	17370	0	1157	10560	1	0.9376	0.9602	5.27/21.64	16.003
	75 %	3	18233	0	140	10714	1	0.9924	0.9952	9.58/21.23	19.066
	50 %	4	18248	0	127	10712	1	0.9931	0.9956	6.23/21.35	18.904
	25 %	7	18271	0	107	10709	1	0.9942	0.9963	6.02/22.22	18.893
Keyframe 2	100 %	3	15813	0	1559	12004	1	0.9103	0.9469	5.01/20.11	16.18
	75 %	3	16882	0	356	12138	1	0.9793	0.9879	5.68/22.74	18.406
	50 %	4	16907	0	331	12138	1	0.9808	0.9887	6.82/21.61	19.37
	25 %	7	17051	0	191	12134	1	0.9889	0.9935	6.51/23.09	18.952
Keyframe 3	100 %	2	14875	0	1140	13104	1	0.9288	0.9609	5.38/18.94	17.299
	75 %	3	16027	0	424	12668	1	0.9742	0.9854	8.49/19.4	17.781
	50 %	4	16386	0	81	12652	1	0.9951	0.9972	6.56/23.7	18.895
	25 %	7	16476	0	23	12620	1	0.9986	0.9992	6.86/23.33	19.098

TABLE 10. The rectangular ROI classification results using the simulation’s keyframe data.

	The distance between the center points [m] (Ratio to the height of the rectangle [%])			
	12m (100%)	9m (75%)	6m (50%)	3m (25%)
Keyframe 1				
	14277 points (49.08%)	14311 points (49.2%)	14282 points (49.1%)	14280 points (49.09%)
Keyframe 2				
	15081 points (51.34%)	15217 points (51.8%)	15228 points (51.84%)	15323 points (52.16%)
Keyframe 3				
	14087 points (48.38%)	14160 points (48.63%)	14146 points (48.58%)	14146 points (48.58%)

ation results through (3) using the four categories. In addition, for the processing time, we used 100 repeated measurements

as before, and derived minimum, maximum, and average processing time. The number of the ROIs, classification results of

TABLE 11. The rectangular ROI classification evaluation metrics and processing time using the simulation's keyframe data.

		The num. of ROI [EA]	TP [EA]	FP [EA]	FN [EA]	TN [EA]	Precision	Recall	Accuracy	Min/Max time [ms]	Average time [ms]
Keyframe 1	100 %	5	14277	0	88	14722	1	0.9939	0.997	5.44/21.27	17.602
	75 %	6	14311	0	51	14725	1	0.9964	0.9982	6.1/22.85	18.907
	50 %	8	14282	0	0	14805	1	1	1	7.68/22.63	20.249
	25 %	15	14280	0	0	14807	1	1	1	7.29/27.17	18.537
Keyframe 2	100 %	5	15073	8	365	13930	0.9995	0.9764	0.9873	5.49/23.21	15.289
	75 %	6	15213	4	153	14006	0.9997	0.99	0.9947	7.84/22.7	19.043
	50 %	8	15219	9	83	14065	0.9994	0.9946	0.9969	6.4/23.54	19.575
	25 %	15	15310	13	0	14053	0.9992	1	0.9996	7.1/27.2	20.073
Keyframe 3	100 %	4	14087	0	22	15010	1	0.9984	0.9992	5.33/21.87	18.212
	75 %	6	14160	0	0	14959	1	1	1	6.14/23.72	18.218
	50 %	8	14146	0	0	14973	1	1	1	5.81/24.1	18.426
	25 %	15	14146	0	0	14973	1	1	1	8.68/25.94	20.977

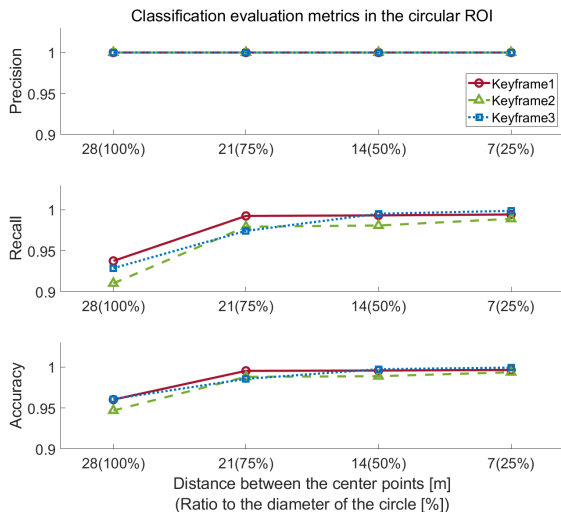


FIGURE 11. The circular ROI classification evaluation using the simulation's keyframe data.

the points, classification evaluation metrics, and processing time, listed in Tables 9 and 11, are summarized and presented as graphs in Figs. 11–14.

The simulation results show that circular and rectangular ROIs have different characteristics depending on whether the target trajectory is straight (Keyframes 1 and 3) or curved (Keyframe 2) like the previous validation, and the distance between the center points of the ROI. The circular ROI is contained within the ground truth ROI for all cases, regardless of the target trajectory (Table 8), and therefore does not fill it up. In addition, the closer the distance between the center points, the more space within the ground truth ROI is filled. Therefore, Table 9 shows that there are zero FP points. The number of FN points reduces as the center points come closer. Therefore, as listed in Table 9, the precision has a value of one

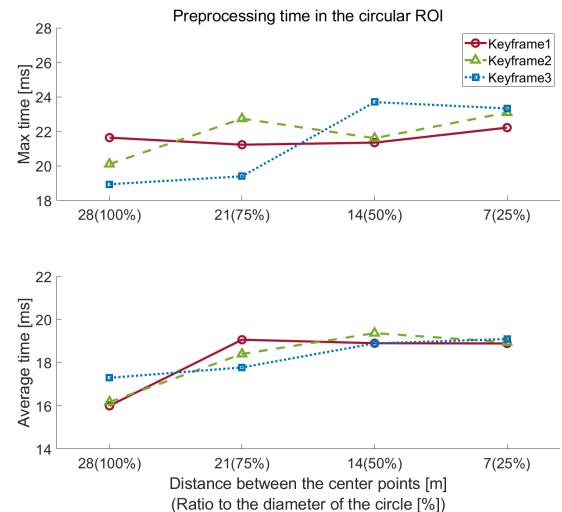


FIGURE 12. The circular ROI processing time using the simulation's keyframe data.

for all cases. The recall and accuracy increase as the distance between the center points decreases. Fig. 11 shows graphically the precision, recall, and accuracy values detailed in Table 9. It is evident that precision always has a value of one, and the recall and accuracy increase as the distance decreases. Because all keyframes had almost the same number of points, there was no significant difference in the average processing time. However, the number of ROIs required increases with the closeness of the center points in a keyframe. Therefore, the average processing time is longer in a keyframe. Table 9 summarizes the minimum, maximum, and average times of the 100 measurements as numerical values. Fig. 12 shows the maximum and average values. Fig. 12 shows that the average processing time is larger when the center points are closer in most keyframes. Furthermore, for all cases, the longest

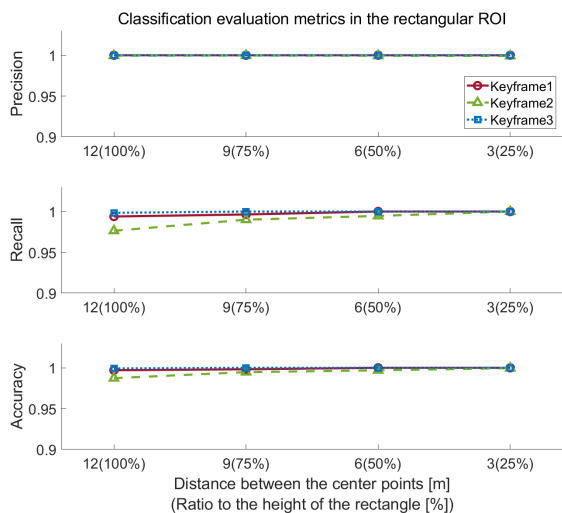


FIGURE 13. The rectangular ROI classification evaluation metrics using the simulation’s keyframe data.

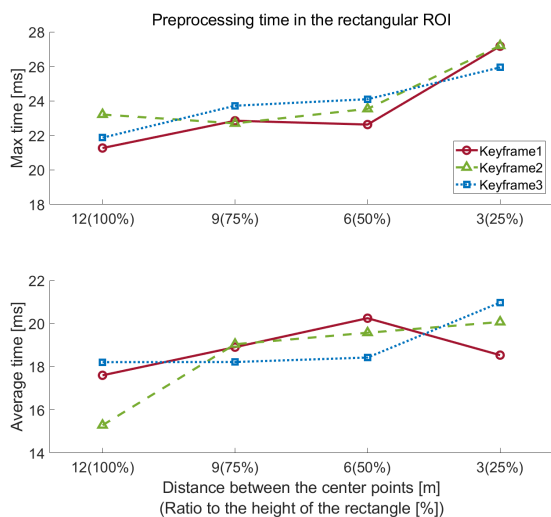


FIGURE 14. The rectangular ROI processing time using the simulation’s keyframe data.

maximum processing time in a single keyframe, in the worst case, was approximately 24 ms.

The rectangular ROI has different characteristics depending on the shape of the target trajectory, unlike the circular ROI. As shown in Table 10, the ROI was almost identical to the ground truth ROI, regardless of the distance between the center points in the straight trajectory, as in Keyframes 1 and 3. However, in curved trajectories, such as Keyframe 2, the further the center points are from each other, the fewer spaces it fills the ground truth ROI, and there is a protruding space as well. In Table 11, the FP points exist only in Keyframe 2, which is a curved trajectory; the number of FN points decreases as the center points come closer. As shown in Table 11, the precision has a value of one only in Keyframes 1 and 3; the recall and accuracy have relatively low values when the center points are further away.

In addition, in Fig. 13, which shows these values graphically, Keyframe 2 has lower values than the rest of the keyframes, but almost the same values as the distance decreases. For the average time, there is no significant difference between keyframes, similar to a circular ROI. However, the processing time normally increases with an increase in the number of ROIs caused by the decreased distance between the center points. In Table 11, the minimum, maximum, and average processing times are summarized for the rectangular ROI. In addition, Fig. 14 shows the maximum and average times. Regarding the average time, most of them took longer when the center points were close together. The longest maximum time in one keyframe was approximately 27 ms when the shortest distance was used.

As the distance between center points shrinks, the circular and rectangular ROIs are more consistent with the ground truth ROI. All classification evaluation metrics have values of 0.99 or higher at the shortest distance. Therefore, we can conclude that these circular and rectangular ROIs exhibit satisfactory performance in terms of the target. Additionally, the circular ROI had 42 FPS, and the rectangular ROI had 37 FPS when the longest maximum time was considered. The proposed methods exhibited faster processing rates than the raw 3D point cloud acquisition rate.

IV. EXPERIMENT

We performed the two experiments in a real-world setting to verify the simulation results of the circular and rectangular ROIs. First, we generated the more complex trajectory in a real-world environment, and we validated the results of the two proposed ROI determination methods on the keyframe data acquired in real-world environments in the same way as in the simulation. Second, to validate dynamic situations in which the robot moves along the given trajectory as well as the static keyframes, we obtained the sequential frame data using the trajectory generated in the previous experiment, and evaluated the processing time of the two ROIs. We also increased the number of points within one point cloud data by accumulating the sequential frame data, and identified the processing times of the two ROI determination methods as the number of points increased. Finally, we conducted a qualitative comparison of these ROI determination methods with the method of the prior study.

A. EXPERIMENT ON THE KEYFRAMES

In the experiment, we set up a more complex trajectory than the simulation (see the solid yellow line in Fig. 15) in an outdoor environment. The same model of mobile robot and LiDAR used in the simulation was used here. Additionally, we used ZED-F9P GPS and VECTORNAV’s VN-100 IMU (Figs. 16 to 17) to obtain the pose information of the robot. We built an Ubuntu 18.04 environment on the NUC processor using an Intel i5-10210U CPU mounted on the mobile robot. The overall control and sensing were performed using the ROS melodic. We controlled the mobile robot at a constant velocity of 1 m/s along the given trajectory and collected the

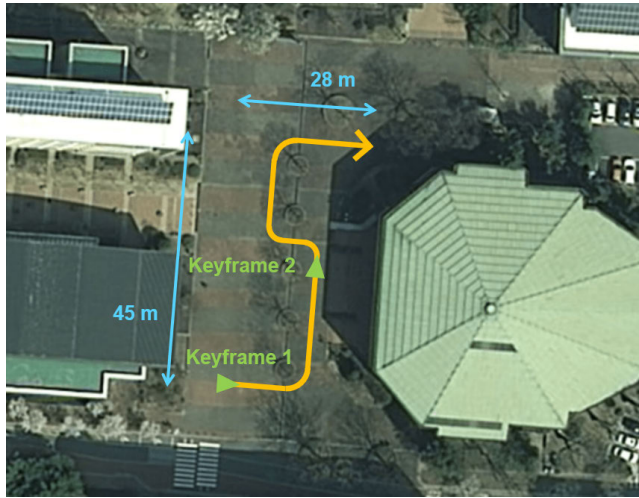


FIGURE 15. The map of the experiment.



FIGURE 17. VECTORNAV's VN-100 IMU [26].



FIGURE 16. ublox's ZED-F9P GPS [25].

TABLE 12. The information of the keyframes used in the experiment.

	Position X [m]	Position Y [m]	Heading θ [rad]
Keyframe 1	90994.33	-44595.87	-0.422
Keyframe 2	91012.87	-44572.53	1.461

pose information at 8 Hz and 62 Hz from the GPS and IMU, respectively. In addition, we obtained the 3D point cloud data at 10 Hz via VLP-16 LiDAR. We obtained raw latitude and longitude data from the GPS and converted them into XY plane coordinates as Table 12.

As shown in Fig. 15 and Table 12, similar to the simulation, we generated two keyframes on the given trajectory and listed the raw 3D point cloud data projected onto the XY plane and the number of those in Table 13. Additionally, we set the parameters of the two ROIs as follows: a diameter of 12 m for the circular ROI, a height of 6 m, and a width of 12 m for the rectangular ROI, and we set a minimum center point distance of 36 m for both ROIs. We set the distances between the center points to the four cases set in the simulation: 25%, 50%, 75%, and 100% of the diameter for the circular ROI and the height of the rectangular ROI. We applied these ROIs to all keyframes and distances to extract the points, as shown in Tables 14–17.

In addition, we conducted a performance evaluation of both ROIs in the same manner as in the simulation. Similar to the solid red lines in Tables 14 and 16, we determine the ground truth ROIs to measure whether the two proposed ROIs have an exact width along the target trajectory. Using this, we classified all points into four categories: TP, FP, FN, and TN. For each of the circular and rectangular ROIs, the classification results for all cases are listed in Tables 14 and 16. The number of points extracted and the ratio of the total number are summarized. Using (3), we calculated the classification evaluation metrics of precision, recall, and accuracy, as we did for the simulation. We detailed the results of the classification evaluation metrics in Tables 15 and 17, and plotted the results in Figs. 18 and 20. Similarly, we measured the processing time in the same manner as in the simulation: minimum, maximum, and average time of 100 measurements. Tables 15 and 17 summarize these time values as numerical values. Figs. 19 and 21 show the maximum and average values, respectively.

Although, unlike in the simulation, the keyframes in the experiment did not have a clear straight or curved target trajectory, their results were almost identical. The results of the circular ROI matched the simulation because it was independent of the shape of the target trajectory. As shown in Table 14, the ROI is contained within the ground truth ROI, and it fills up as the distance between the center points decreases. Table 15 shows that the number of FP points is zero and that the number of FN points decreases when the distance becomes smaller. Therefore, the precision has a value of one for all cases, and the recall and accuracy increase as the distance decreases. Fig. 18 shows the constant precision with one and the recall and accuracy that increase with a decrease in distance. The average processing time in one keyframe was similar between the keyframes, and as the number of required ROIs increased, a higher value was required. As shown in Fig. 19, the average time tends to increase as the distance decreases. In addition, the longest maximum time in a single keyframe was approximately 8 ms at a distance of 50 %.

Furthermore, the rectangular ROI has almost identical ground truth ROI as the straight trajectory but differs in the curved trajectory. Therefore, our findings are affected because the ROI shows different results for relatively curved Keyframes 1 and 2. Table 17 shows that more FN points are found in Keyframe 2, which has greater curves, and their numbers decrease as the center points come closer.

TABLE 13. The real point cloud data of the keyframes in the experiment.

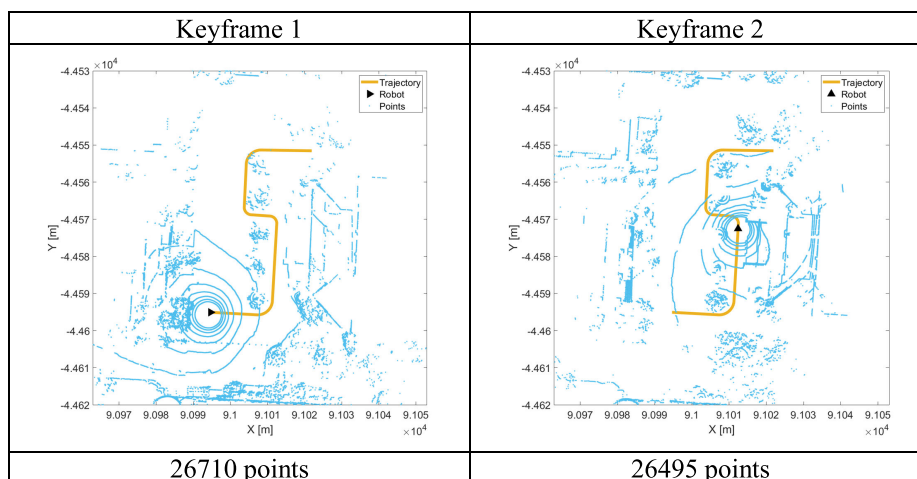


TABLE 14. The circular ROI classification results using the experiment's keyframe data.

		The distance between the center points [m] (Ratio to the diameter of the circle [%])			
		12m (100%)	9m (75%)	6m (50%)	3m (25%)
Keyframe 1					
		8379 points (31.37%)	8733 points (32.7%)	8991 points (33.66%)	9093 points (34.04%)
Keyframe 2					
		10520 points (39.71%)	10681 points (40.31%)	10968 points (41.4%)	11238 points (42.42%)

In addition, the precision between the keyframes is similar, but the recall and accuracy are lower in Keyframe 2 when the distance is greater. Fig. 20 shows the differences between the two keyframes. Both keyframes have similar values close to one in precision, but Keyframe 2 has lower values in recall and accuracy when the distances are greater. In the case of the average processing time in one keyframe, there is no difference between the keyframes as in the circular ROI, but less distance makes the value greater. As shown in Fig. 21, for both keyframes, the average time required increased as the distance decreased. In addition, the longest maximum processing time in one keyframe was approximately 11 ms in the case of the shortest distance.

The values of the classification evaluation metrics are 0.99 or higher when the distance between the center points is the shortest. Both ROIs show quicker than the data; the circular and the rectangular ROIs exhibit speeds of 125 FPS and 91 FPS, respectively, for the longest maximum time. Therefore, we can conclude that the proposed methods provide high performance.

B. EXPERIMENT ON THE SEQUENTIAL FRAMES

We obtained 1327 sequential 3D point cloud frames by moving the robot along the trajectory with the same settings as the experiment on the keyframes, and utilized them to derive two results. At this time, we fixed the distance between the

TABLE 15. The circular ROI classification evaluation metrics and processing time using the experiment’s keyframe data.

		The num. of ROI [EA]	TP [EA]	FP [EA]	FN [EA]	TN [EA]	Precision	Recall	Accuracy	Min/Max time [ms]	Average time [ms]
Keyframe 1	100%	4	8379	0	760	17571	1	0.9168	0.9715	4.92 / 5.91	5.235
	75%	5	8733	0	406	17571	1	0.9556	0.9848	5 / 6.89	5.289
	50%	7	8991	0	146	17573	1	0.984	0.9945	5.21 / 7.98	5.408
	25%	13	9093	0	32	17585	1	0.9965	0.9988	5.81 / 6.76	6.039
Keyframe 2	100%	4	10520	0	852	15123	1	0.9251	0.9678	4.91 / 5.8	5.148
	75%	5	10681	0	691	15123	1	0.9392	0.9739	5.01 / 6.73	5.194
	50%	7	10968	0	399	15128	1	0.9649	0.9849	5.16 / 6.57	5.429
	25%	13	11238	0	111	15146	1	0.9902	0.9958	5.69 / 6.48	5.825

TABLE 16. The rectangular ROI classification results using the experiment’s keyframe data.

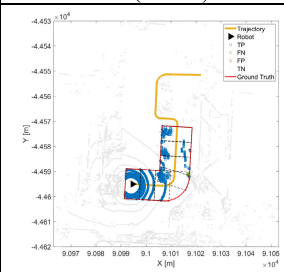
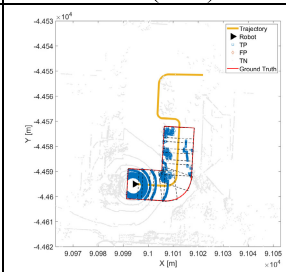
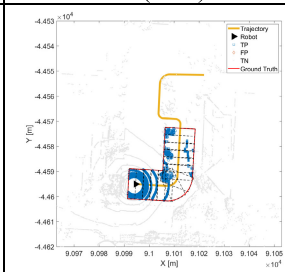
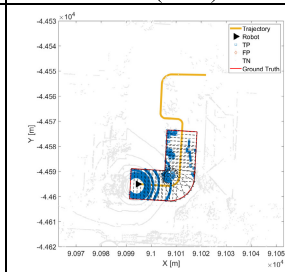
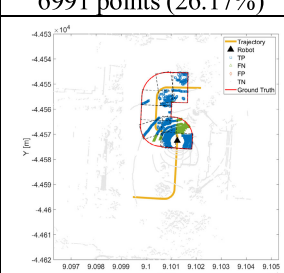
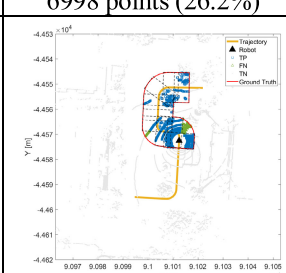
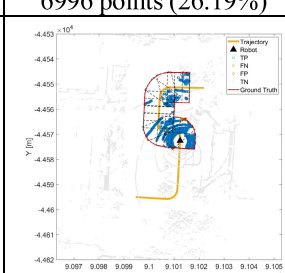
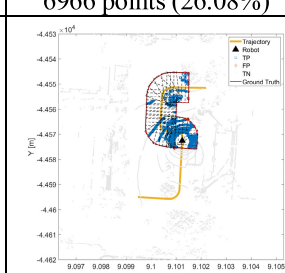
		The distance between the center points [m] (Ratio to the height of the rectangle [%])			
		6m (100%)	4.5m (75%)	3m (50%)	1.5m (25%)
Keyframe 1					
		6991 points (26.17%)	6998 points (26.2%)	6996 points (26.19%)	6966 points (26.08%)
Keyframe 2					
		8537 points (32.22%)	9174 points (34.63%)	9885 points (37.31%)	9856 points (37.2%)

TABLE 17. The rectangular ROI classification evaluation metrics and processing time using the experiment’s keyframe data.

		The num. of ROI [EA]	TP [EA]	FP [EA]	FN [EA]	TN [EA]	Precision	Recall	Accuracy	Min/Max time [ms]	Average time [ms]
Keyframe 1	100%	7	6983	8	14	19705	0.9989	0.998	0.9992	5.35 / 6.66	5.559
	75%	9	6994	4	0	19712	0.9994	1	0.9999	5.71 / 7.09	5.97
	50%	13	6992	4	0	19714	0.9994	1	0.9999	6.35 / 7.37	6.703
	25%	25	6960	6	0	19744	0.9991	1	0.9998	8.3 / 10.64	8.634
Keyframe 2	100%	7	8533	4	1357	16601	0.9995	0.8628	0.9486	5.66 / 7.66	5.927
	75%	9	9174	0	708	16613	1	0.9284	0.9733	5.87 / 6.61	6.139
	50%	13	9826	59	45	16565	0.994	0.9954	0.9961	6.46 / 7.28	6.738
	25%	25	9804	52	0	16639	0.9947	1	0.998	8.1 / 10.27	8.454

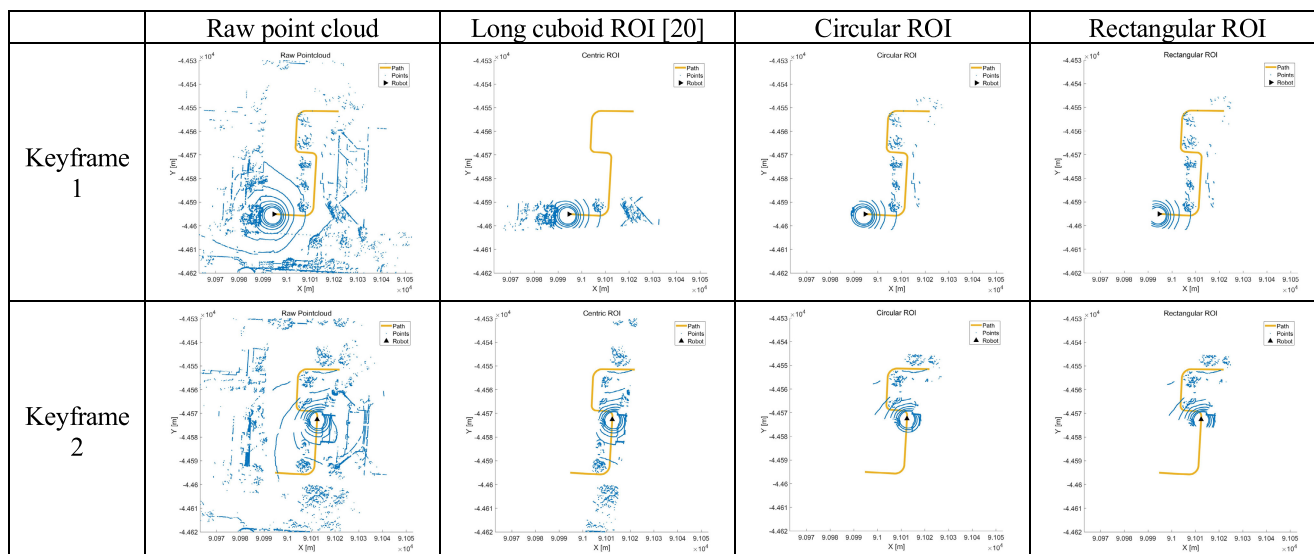
TABLE 18. The circular and rectangular ROIs processing time using the experiment’s sequential frame data.

	Min time [ms]	Max time [ms]	Average time [ms]
Point cloud cycle	94.895	106.812	100.856
Circular ROI processing time	4.718	9.407	5.555
Rectangular ROI processing time	6.434	13.646	7.928

TABLE 19. The circular and rectangular ROIs results using the experiment’s accumulated frame data.

	The cumulative num. of the sequential frames [EA]	The num. of frames accumulated [EA]	Average num. of points [EA]	Min time [ms]	Max time [ms]	Average time [ms]
Circular ROI	1	1327	26237.45	4.718	9.407	5.555
	5	265	131179.74	25.538	28.760	27.239
	10	132	262346.83	51.005	74.361	54.731
	25	53	655898.68	126.987	149.783	137.898
	50	26	1311490.08	259.137	299.993	278.35
	75	17	1966791.82	424.148	481.319	437.885
	100	13	2622980.15	560.974	650.781	603.667
Rectangular ROI	1	1327	26237.45	6.434	13.646	7.928
	5	265	131179.74	37.524	45.068	40.205
	10	132	262346.83	75.491	90.885	80.811
	25	53	655898.68	186.356	237.293	200.883
	50	26	1311490.08	384.069	442.899	405.898
	75	17	1966791.82	616.388	658.933	629.026
	100	13	2622980.15	816.388	895.131	853.505

TABLE 20. The qualitative comparison with the existing study using the experiment’s keyframe data.



center points at the smallest 25% ratio that derived the best classification evaluation results in the previous experiment. First, we compared the 3D point cloud data cycle with the processing time of the two proposed ROI determination methods for the sequential frames. For the total of 1327 frames, we measured the data cycle and processing time of the two

ROIs, and showed the results in Fig. 22 and summarized the calculated minimum, maximum and average time in Table 18. Looking at the average time results in Table 18, the point cloud data were obtained for the 10 Hz we set, and the processing time of the two ROIs was similar to that of the previous experiment on the keyframes. In addition, as shown

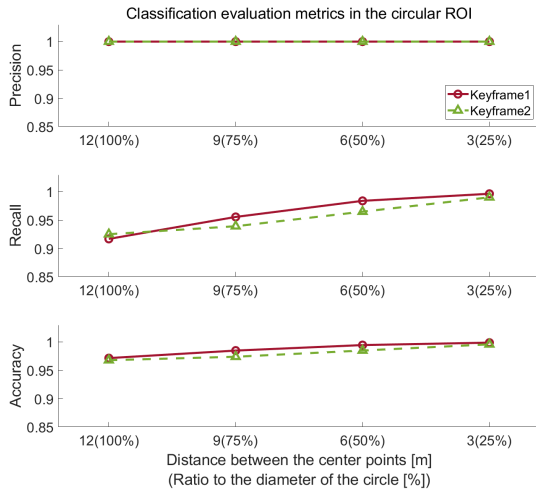


FIGURE 18. The circular ROI classification evaluation using the experiment's keyframe data.

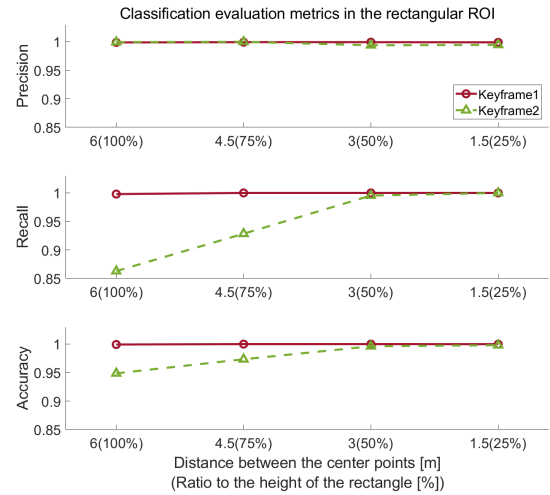


FIGURE 20. The rectangular ROI classification evaluation metrics using the experiment's keyframe data.

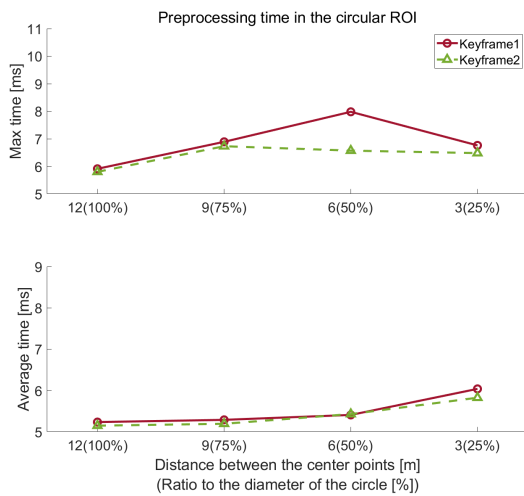


FIGURE 19. The circular ROI processing time using the experiment's keyframe data.

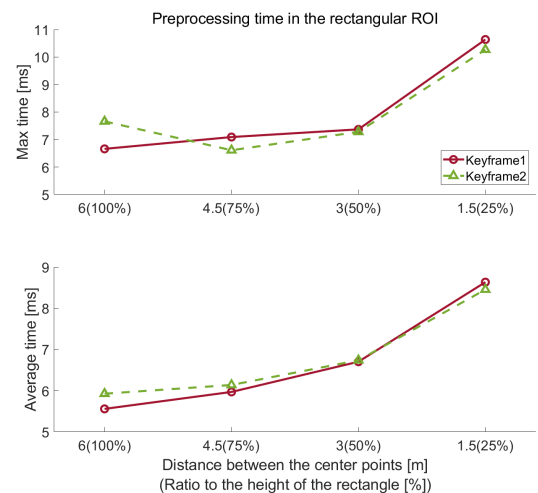


FIGURE 21. The rectangular ROI processing time using the experiment's keyframe data.

in Fig. 22, which is the results of the entire sequential frames, both ROIs have almost uniform processing time and much smaller values than the data cycle. This verifies that the processing speed of the two ROIs is faster than the data acquisition rate (on average, approximately 18 and 13 times, respectively), not only in the static keyframes but also in the sequential frames where the robot moves dynamically.

Second, we confirmed the processing time of the proposed ROI determination methods for point cloud data with varying numbers of points in one frame by accumulating the previously acquired sequential data. The number of points within the point cloud is determined by relying on sensors. However, we adopted the cumulative method to check the tendency of the processing time of the two ROIs for the different number of points. We conducted the experiment by setting the cumulative number of frames to a total of seven; 1, 5, 10, 25, 50, 75, and 100. At this time, as explained in II-A, we transformed the accumulated data into the global coordinate based on robot pose information when reaching

the cumulative target number, and extracted points along the trajectory by applying two ROIs. As a result, the number of frames accumulated, the average number of points, and the minimum/maximum/average processing time for each case of the two ROIs are summarized in Table 19. Here, the data with the cumulative number of 1 corresponds to the previous sequential frames, and the experiment was conducted by accumulating this data. Also, Fig. 23 graphs the minimum/maximum/average time of the two ROIs according to the number of each cumulative frame and the average number of points. In the graph, the circle and triangle represent the average time, and the vertical bar represents the minimum/maximum time. Looking at the results, it was confirmed that as the number of points increased by increasing the cumulative amount of the sequential frame, the processing time of the two ROIs increased almost linearly. Therefore, these results allow us to infer the tendency that the processing time is required in proportion to the number of points in one

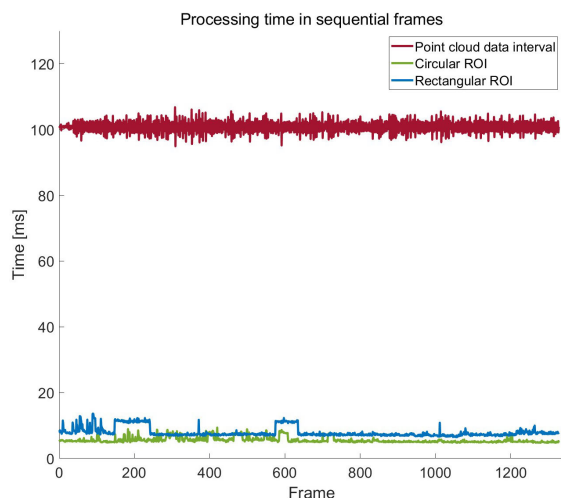


FIGURE 22. The circular and rectangular ROIs processing time and point cloud data interval using the experiment’s sequential frame data.

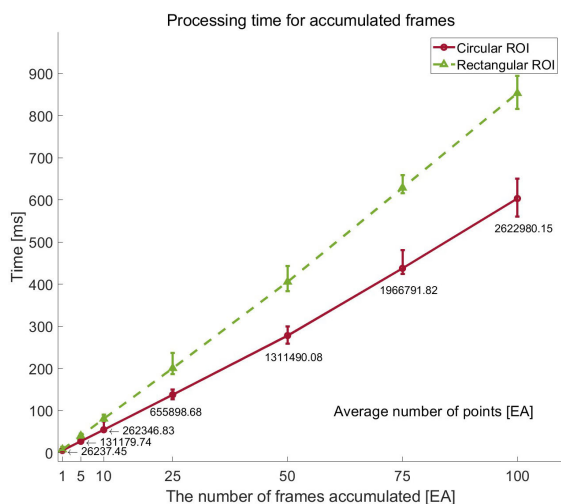


FIGURE 23. The circular and rectangular ROIs processing time and average number of points using the experiment’s accumulated frame data.

point cloud data. In addition, through this tendency, we can find the maximum number of points that can be processed within the data acquisition cycle.

C. QUALITATIVE COMPARISON WITH PREVIOUS STUDY

We compared the methods of this study with the method of the existing study, long cuboid ROI determination method [20], using the same keyframe set in the previous experiment. Because the results of each method varied greatly depending on the parameter setting, the quantitative comparison was difficult; therefore, we only conducted qualitative comparisons.

In the case of the long cuboid ROI determination method, an ROI with values of 10 m in height, a width of 12 m, and a length of 100 m was determined around the robot. For the ROI determination methods proposed in this study, maintaining the experiment setting, the ROI was determined to the end of the trajectory, and the distance between the center points was

fixed to the smallest value. We summarized the results of all methods in Table 20.

The comparison results show that the long cuboid ROI determination method can reduce the number of points by extracting only the points around the robot; however, there is a limit to the response speeds of the target task because the ROI does not precisely match the trajectory. By contrast, because the circular and rectangular ROIs are determined based on the target trajectory, only points close to that trajectory can be efficiently extracted while maintaining the resolution of the raw data.

V. CONCLUSION

This paper proposes 3D point cloud reduction methods that extract only important data close to the target trajectory. These methods sequentially determine the circular and rectangular ROIs along the trajectory and only take the points within these two ROIs. We conducted the verification of the two proposed ROI determination methods through the three types of data. Through the artificial frames, we confirm that the distance parameter between the center points causes the trade-off relationship between the classification evaluation results and the processing time of the two ROIs, so that the user should set the distance parameter with proper consideration of the relationship between the two results. In addition, through the keyframes and sequential frames, we confirm that when the distance parameter is 25% of the diameter or height in circular and rectangular ROIs, both in static and dynamic situations, the classification evaluation results are close to 1, and the processing time is much shorter than the data cycle. There are two main contributions to these methods. First, we can efficiently extract only the important points near the trajectory while maintaining a high resolution of the raw data. Second, regardless of the complexity of the trajectory, we can determine the exact matching ROI.

The proposed methods can be used in various scenarios. First, in an autonomous driving environment, the vehicle can only obtain important 3D point cloud data near the trajectory to the destination rather than data on distant vehicles and pedestrians in the opposite lane. In addition, a mobile robot in a factory, such as the simulation in this study, can move quickly while avoiding workers and other robots by utilizing only data on the target trajectory. Finally, outdoor delivery robots that have been developed recently can complete the target task by focusing on the data near the target trajectory, although they operate in a vast and complex dynamic environment.

However, the methods proposed in this study have limitations. Since these methods determine the ROI along the robot’s target trajectory, it is essential and is planned first. In the future, we will conduct additional studies on adjusting the width efficiently according to the robot’s velocity rather than the fixed width for the corresponding ROI determination methods. In addition, we plan to verify our results through experiments using various sensors and robots.

REFERENCES

- [1] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 1–4.
- [2] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nüchter, and J. Wang, "3D LiDAR point cloud based intersection recognition for autonomous driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2012, pp. 456–461.
- [3] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR point cloud generator: From a virtual world to autonomous driving," in *Proc. ACM Int. Conf. Multimedia Retr.*, Yokohama, Japan, Jun. 2018, pp. 458–464.
- [4] M. Yuda, Z. Xiangjun, S. Weiming, and L. Shaofeng, "Target accurate positioning based on the point cloud created by stereo vision," in *Proc. 23rd Int. Conf. Mechatronics Mach. Vis. Pract. (M2VIP)*, Nanjing, China, Nov. 2016, pp. 1–5.
- [5] T. Hoegg, D. Lefloch, and A. Kolb, "Time-of-flight camera based 3D point cloud reconstruction of a car," *Comput. Ind.*, vol. 64, no. 9, pp. 1099–1114, Dec. 2013.
- [6] R. Y. Takimoto, M. D. S. G. Tsuzuki, R. Vogelaar, T. D. C. Martins, A. K. Sato, Y. Iwao, T. Gotoh, and S. Kagei, "3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor," *Mechatronics*, vol. 35, pp. 11–22, May 2016.
- [7] L. Mu, P. Yao, Y. Zheng, K. Chen, F. Wang, and N. Qi, "Research on SLAM algorithm of mobile robot based on the fusion of 2D LiDAR and depth camera," *IEEE Access*, vol. 8, pp. 157628–157642, 2020.
- [8] K. Lai and D. Fox, "Object recognition in 3D point clouds using web data and domain adaptation," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1019–1037, 2010.
- [9] E. Che, J. Jung, and M. J. Olsen, "Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review," *Sensors*, vol. 19, no. 4, pp. 810–1–810–42, Feb. 2019.
- [10] Y. Wu, S. Zhang, H. Ogai, H. Inujima, and S. Tateno, "Realtime single-shot refinement neural network with adaptive receptive field for 3D object detection from LiDAR point cloud," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24505–24519, Nov. 2021.
- [11] X. T. Nguyen, K. T. Nguyen, H. J. Lee, and H. Kim, "ROI-based LiDAR sampling algorithm in on-road environment for autonomous driving," *IEEE Access*, vol. 7, pp. 90246–90253, 2019.
- [12] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LiDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4901–4913, May 2020.
- [13] F. Gao and S. Shen, "Online quadrotor trajectory generation and autonomous navigation on point clouds," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Lausanne, Switzerland, Oct. 2016, pp. 139–146.
- [14] Y. Fan, Q. Zhang, S. Liu, Y. Tang, X. Jing, J. Yao, and H. Han, "Semantic SLAM with more accurate point cloud map in dynamic environments," *IEEE Access*, vol. 8, pp. 112237–112252, 2020.
- [15] C. Moreno and M. Li, "A comparative study of filtering methods for point clouds in real-time video streaming," in *Proc. World Congr. Eng. Comput. Sci.*, San Francisco, CA, USA, 2016, pp. 388–393.
- [16] A. Hacinecipoglu, E. I. Konukseven, and A. B. Koku, "Pose invariant people detection in point clouds for mobile robots," *Int. J. Mech. Eng. Robot. Res.*, vol. 9, no. 5, pp. 709–715, May 2020.
- [17] Y. Xu, X. Tong, and U. Stilla, "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry," *Autom. Construct.*, vol. 126, Jun. 2021, Art. no. 103675.
- [18] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021.
- [19] K. Jo, M. Lee, and M. Sunwoo, "Track fusion and behavioral reasoning for moving vehicles based on curvilinear coordinates of roadway geometries," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 3068–3075, Sep. 2018.
- [20] S. Mou, Y. Chang, W. Wang, and D. Zhao, "An optimal LiDAR configuration approach for self-driving cars," 2018, *arXiv:1805.07843*.
- [21] Ahtsan. *Aws-Robomaker-Small-Warehouse-World*. Accessed: Dec. 7, 2022. [Online]. Available: <https://github.com/aws-robotics/aws-robomaker-small-warehouse-world>
- [22] AgileX Robotics Co. *Scout 2.0 E-Book*. Accessed: Dec. 7, 2022. [Online]. Available: https://cdn.shopify.com/s/files/1/0551/0630/6141/files/SCOUT_2.0_E-BOOK.pdf?v=1637235031
- [23] Velodyne LiDAR. *VLP-16 User Manual*. Accessed: Aug. 2, 2022. [Online]. Available: <https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf>
- [24] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process.*, vol. 5, no. 2, pp. 1–11, 2015.
- [25] U-Blox. *ZED-F9P Integration Manual*. Accessed: Dec. 7, 2022. [Online]. Available: https://www.u-blox.com/en/ubx-viewer/view/ZED-F9P_IntegrationManual_UBX-18010802?url=https%3A%2F%2Fwww.u-blox.com%2Fsites%2Fdefault%2Ffiles%2FZED-F9P_IntegrationManual_UBX-18010802.pdf
- [26] VectorNav Technologies. *VN100 Manual*. Accessed: Dec. 7, 2022. [Online]. Available: <https://www.eol.ucar.edu/system/files/VN100manual.pdf>



JONG HOON PARK received the B.S. degree in mechanical and information engineering from the University of Seoul, Seoul, South Korea, in 2021, where he is currently pursuing the M.S. degree in mechanical and information engineering. His research interests include motion planning and control of mobile manipulator.



YE EUN LIM is currently pursuing the degree in mechanical and information engineering with the University of Seoul. Her research interests include autonomous mobility, robot vision, and reinforcement learning.



JUNG HYUN CHOI is currently pursuing the degree in mechanical and information engineering with the University of Seoul. His research interests include autonomous mobility, control, and quadruped robot.



MYUN JOONG HWANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2001, 2003, and 2007, respectively. He has been an Associate Professor with the Department of Mechanical and Information Engineering, University of Seoul, South Korea, since 2021. He was a Research Associate at the Mechanical Engineering Research Institute, KAIST, in 2007. From 2008 to 2009, he was a Research Associate with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, USA. He was a Senior Research Engineer at the Manufacturing Technology Center, Samsung Electronics Company Ltd., Suwon, South Korea, from 2010 to 2013. He was an Assistant Professor at the School of Mechanical and Automotive Engineering, Halla University, Wonju, South Korea, from 2013 to 2015, and an Assistant Professor/an Associate Professor with the Department of Mechanical Engineering, Korea National University of Transportation, Chungju, South Korea, from 2015 to 2021. His research interests include robot motion planning and control, manipulation, autonomous driving, and robot vision.

• • •