SLAM-based Mapping in Truck-and-Robot System for Last-Mile Delivery Automation

Ryo Nakamura
Graduate School of Science and Engineering
Doshisha University
Kyotanabe, Japan
e-mail: ctwh0151@mail4.doshisha.ac.jp

Takeshi Kambe, Masafumi Hashimoto,
Kazuhiko Takahashi
Faculty of Science and Engineering
Doshisha University
Kyotanabe, Japan
e-mail: {mhashimo, katakaha}@mail.doshisha.ac.jp

***Abstract*—This paper presents a Simultaneous Localization And Mapping (SLAM)-based mapping method for last-mile delivery automation using a scanning Light Detection And Ranging sensor (LiDAR) mounted on a quadruped robot. Distortion in scan data from the LiDAR, caused by the swinging motion of the robot, is corrected by estimating the robot's pose (three-dimensional positions and attitude angles) in a period shorter than the LiDAR scan period using an extended Kalman filter. LiDAR-scan data related to stationary objects are detected from the corrected scan data using an occupancy grid method. Local maps in small areas where robots deliver goods to customers are built using normal distributions transforms and Graph SLAM. A feature-based loop detection is also performed using surface features and point feature histograms. The local maps are corrected in the Graph SLAM framework using the scan data from LiDAR mounted on a truck stopping at robot depots. Experimental results obtained in our university campus demonstrate the effectiveness of the presented method.**

Keywords—LiDAR; NDT Graph SLAM; map building; loop detection; quadruped robot; delivery automation.

I. INTRODUCTION

Recently, last-mile delivery automation using wheeled and legged robots has progressed due to increased e-commerce and demand for contactless delivery during the COVID-19 pandemic [1][2]. Delivery robots are designed to move short distances at pedestrian speed. Owing to their low speed and limited range, delivery robots are usually combined with trucks to enable a fast and efficient delivery process [3][4]. As shown in Figure 1, a truck transports delivery goods with robots and releases the robots at dedicated drop-off locations (robot depots). The robots deliver goods to customers and return to the robot depots by themselves.

In such truck-and-robot delivery systems, map building (mapping) and map-matching-based self-localization using built maps are important technologies for autonomous navigation of delivery robots [5]. In the domain of mobile robotics and Intelligent Transportation Systems (ITS), many related studies using cameras and Light Detection And Ranging sensors (LiDARs) have been presented [6]–[8]. Mobile mapping systems are typically used to build High-Definition (HD) maps for autonomous driving and advanced driver assistant systems in wide road environments, such as highways and motorways. In truck-and-robot delivery systems, autonomous driving and pose estimation of trucks moving in wide road environments can be performed using HD maps. However, because HD maps building by mobile mapping systems incur high cost, Simultaneous Localization And Mapping (SLAM)-based mapping has been proposed as an efficient method for mapping narrow residential environments, in which robots deliver goods to customers.

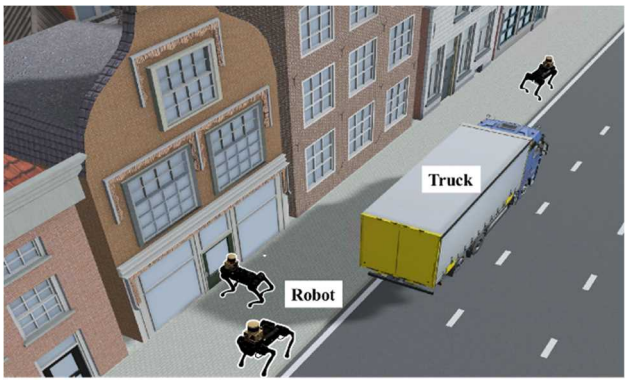


Figure 1. Image of truck-and-robot delivery system.

In this paper, we focus on LiDAR SLAM-based mapping. We previously presented mapping methods using LiDAR mounted on cars, motorcycles, and driver's helmets based on Normal Distributions Transforms (NDT) SLAM [9]–[11] to build a three-dimensional (3D) point cloud map in community road environments.

To build 3D point cloud maps by robot-mounted LiDAR using scan matching based-SLAM, such as NDT SLAM and iterative closest point SLAM, the scan data captured in the LiDAR coordinate frame are mapped onto the world coordinate frame using the self-pose (position and attitude angle) of a robot. Mechanical LiDARs, where laser beams are scanned in omnidirection (rotation of 360° of the laser beams in the horizontal direction), are typically used for LiDAR-based mapping. Hence, the complete data within one scan (one rotation of the laser beams in the horizontal direction) cannot be acquired simultaneously when a robot is moving and swinging. Therefore, if such data are transformed based on the robot's pose at the same time, distortion appears in the mapping results.

To reduce the distortion in scan data, many methods for distortion correction have been presented using linear interpolation and its variants [12][13]. In our previous work,

 ISBN: 978-1-68558-089-6

a Kalman filter-based method was presented [10][11]. Because Kalman filter-based localization is widely used in the fields of mobile robotics, distortion correction in a Kalman filter framework can be easily incorporated in the self-localization system of a robot.

Scan matching-based SLAM causes a drift (degradation of accuracy over time); to reduce the drift, Graph SLAM is typically used in conjunction with scan matching-based SLAM. In Graph SLAM, the detection of revisit places (called loops) is an important issue, and many methods for loop detection have been presented [14][15]. In our previous work, a detection method using surface features and matching distance indicators was presented [9]. However, some improvements are required to reduce missed and false detection of loops.

This paper presents a LiDAR SLAM-based mapping method for truck-and-robot delivery systems. The LiDAR SLAM-based mapping method involves integrating components that we previously proposed [9]–[11]: distortion correction of LiDAR scan data, extraction of scan data related to stationary objects from the entire corrected LiDAR scan data, and point cloud mapping based on NDT Graph SLAM. Another contribution of this paper is to improve the performance of loop detection in our previous Graph SLAM by introducing Fast Point Feature Histograms (FPFH) [16]. In addition, the mapping accuracy of robot-mounted LiDAR is improved using scan data from truck-mounted LiDAR.

The rest of this paper is organized as follows. Section II describes the experimental system. Section III explains the method of map building and correction, and Section IV presents the method of loop detection. Section V presents experimental results to verify the proposed method, followed by the conclusions in Section VI.

II. Experimental System

Figure 2 shows an overview of a quadruped robot (Unitree A1). A scanning 16-layer LiDAR (Velodyne VLP-16) and an Inertial Measurement Unit (IMU, MTi-300) are mounted on the upper part of the robot. The maximum range of the LiDAR is 70 m, the horizontal viewing angle is 360° with a resolution of 0.2°, and the vertical viewing angle is 30° with a resolution of 2°. The LiDAR provides 384 measurements (the object's 3D position and reflection intensity) every 1.33 ms (at 4.8° horizontal angle increments).

Figure 2. Overview of experimental quadruped robot.

The time that the LiDAR beam takes to complete one rotation (360°) in the horizontal direction is 100 ms, and 30,000 measurements are obtained in one rotation.

The IMU provides attitude angles (roll and pitch angles) and angular velocities (roll, pitch, and yaw velocities) every 10 ms with an attitude angle error of ±0.3° (typ.) and an angular velocity error of ±0.2 °/s (typ.).

Meanwhile, a scanning 32-layer LiDAR (Velodyne HDL-32) is used as a truck-mounted LiDAR. The maximum range of the LiDAR is 70 m, the horizontal viewing angle is 360° with a resolution of 0.16°, and the vertical viewing angle is 41.34° with a resolution of 1.33°. The time that the LiDAR beam takes to complete one rotation (360°) in the horizontal direction is 100 ms, and 70,000 measurements are obtained in one rotation.

III. Map Building and Correction

A. Local Map Building by Robot-Mounted LiDAR

The captured scan data from the robot-mounted LiDAR in a single scan are mapped onto a 3D grid map (voxel map) represented in the LiDAR coordinate frame Σ_b attached to the LiDAR. A voxel grid filter is applied to downsize the scan data. The block used for the voxel grid filter is a cube with a side length of 0.2 m.

In a world coordinate frame Σ_w, a voxel map with a voxel size of 1 m is used for NDT scan matching [17]. For the *i*-th (i = 1, 2, …n) measurement in the scan data, the position vector in Σ_b is denoted as $\boldsymbol{p}_{bi}$ and that in Σ_w as $\boldsymbol{p}_i$. The following relation is obtained:

$$\begin{pmatrix} \boldsymbol{p}_i \\ 1 \end{pmatrix} = \boldsymbol{T}(\boldsymbol{x}) \begin{pmatrix} \boldsymbol{p}_{bi} \\ 1 \end{pmatrix} \tag{1}$$

where $\boldsymbol{x} = (x, y, z, \phi, \theta, \psi)^T$ denotes the robot's pose. $(x, y, z)^T$ and $(\phi, \theta, \psi)^T$ denote the 3D position and attitude angle (roll, pitch, and yaw angles) of the robot, respectively, in Σ_w. $\boldsymbol{T}(\boldsymbol{x})$ denotes the homogeneous transformation matrix:

The scan data obtained at the current time step t (t = 0, 1, 2, …) are called the new input scan, and the scan data obtained in the previous time step, i.e., before $(t-1)$, are called the reference scan (local map). The robot pose at t is determined by matching the new input scan at t with the reference scan data obtained before $(t-1)$. The robot pose is used for coordinate transform using (1). The new input scan can then be mapped to Σ_w, and the local map is updated.

NDT SLAM based on NDT scan matching is performed by mapping LiDAR scan data captured in Σ_b onto Σ_w using the self-pose information of the robot. The LiDAR obtains range measurements by scanning laser beams. Thus, when a robot moves and swings, the complete scan data cannot be acquired in a single scan (LiDAR beam rotation of 360° in a horizontal plane) simultaneously. Therefore, if the entire scan data obtained within one scan are mapped onto Σ_w using robot-pose information at a single point in time, distortion arises in mapping results.

The distortion in the scan data from the LiDAR is corrected by estimating the robot's pose in a period of 1.327

 ISBN: 978-1-68558-089-6

ms, which is shorter than the LiDAR scan period of 100 ms. The extended Kalman filter-based algorithm [11] is applied to distortion correction based on information from NDT SLAM and an IMU.

Corrected scan data relating to road surfaces are removed using a rule-based method [11], and scan data relating to objects are mapped onto the grid map (cell size of 0.3 m in this study). Scan data relating to moving objects (called moving scan data), such as cars and pedestrians, are removed using an occupancy grid method, and those relating to stationary objects (stationary scan data) are then extracted. The stationary scan data are used for NDT SLAM-based mapping.

NDT SLAM degrades mapping accuracy over time due to accumulation errors. To reduce the error, Graph SLAM is employed. The robot poses, which are calculated by NDT SLAM every 100 ms (LiDAR scan period), are mapped onto a pose graph, as depicted in Figure 3. When revisit places (loops), where the robot has already visited places during map building, are detected using a method described in Section IV, the current robot's pose relative to its pose at the revisit node is set to the pose graph as a loop constraint (blue arrow in Figure 3). The objective function of (2) is then minimized to improve the accuracy of the map built by NDT SLAM:

$$J(\boldsymbol{\chi}) = \sum_i \{(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) - \boldsymbol{\delta}_{i+1,i}\}^T \boldsymbol{\Omega}^{pose} \{(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) - \boldsymbol{\delta}_{i+1,i}\} + \sum_{x_A, x_B \in \text{loop}} \{(\boldsymbol{x}_B - \boldsymbol{x}_A) - \boldsymbol{\delta}_{A,B}\}^T \boldsymbol{\Omega}^{loop} \{(\boldsymbol{x}_B - \boldsymbol{x}_A) - \boldsymbol{\delta}_{A,B}\} \quad (2)$$

where the first and second terms on the right side indicate the constraints on NDT SLAM and loop, respectively. $\boldsymbol{\chi} = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_i^T, \cdots)^T$. $\boldsymbol{x}_i$ denotes the robot's pose at the i-th time step. $\boldsymbol{\delta}_{i+1,i}$ denotes the relative pose of the robot between the i-th and (i+1)th time steps, which is calculated from NDT SLAM. $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ denote the robot's poses at the revisit and current nodes, respectively. $\boldsymbol{\delta}_{A,B}$ denotes the relative pose of the robot at the two nodes, which is calculated from the LiDAR scan data using NDT scan matching. $\boldsymbol{\Omega}^{pose}$ and $\boldsymbol{\Omega}^{loop}$ denote the information matrices.

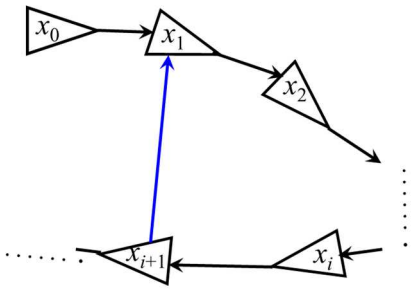


Figure 3. Pose graph for map building. The robot's poses are represented as graph nodes (black triangles), and relative poses between two neighboring nodes are represented as graph edges (black arrows).

B. Map Correction by Truck-Mounted LiDAR

When the robot returns to the robot depot, the local map built by the robot is corrected using the LiDAR scan data captured by the truck-mounted LiDAR. Such map correction is performed in Graph SLAM framework by the following steps:

Step 1: Mapping by truck-mounted LiDAR; the map is built using the truck-mounted LiDAR at the robot depot, and the truck poses, obtained by the map-matching method using an HD map, are mapped onto a pose graph, as depicted in Figure 4;

Step 2: Encounter node detection; nodes, where the robot encounters the truck are detected in the pose graph;

Step 3: Relative pose estimation; the robot's poses relative to the truck at encounter nodes are estimated from scan data captured by the truck and robot-mounted LiDARs using NDT scan matching;

Step 4: Map correction; the local map built by the robot is corrected using pose graph optimization.

The relative poses of the robot at the encounter nodes are set to the pose graph as the loop constraint (red arrow in Figure 4). The following objective function is then minimized to correct the local map:

$$J(\boldsymbol{\chi}') = J(\boldsymbol{\chi}) + \sum_{x^*, x_A \in \text{loop}} \{(\boldsymbol{x}_A - \boldsymbol{x}^*) - \boldsymbol{\delta}_A\}^T \boldsymbol{\Omega}_A^{loop} \{(\boldsymbol{x}_A - \boldsymbol{x}^*) - \boldsymbol{\delta}_A\} + (\boldsymbol{x}^* - \boldsymbol{\delta}^*)^T \boldsymbol{\Omega}^* (\boldsymbol{x}^* - \boldsymbol{\delta}^*) \quad (3)$$

where $\boldsymbol{\chi}' = (\boldsymbol{x}^{*T}, \boldsymbol{\chi}^T)^T$ and $\boldsymbol{\chi} = (\boldsymbol{x}_0^T, \boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_i^T, \cdots)^T$. $\boldsymbol{x}^*$ denotes the truck pose, and $\boldsymbol{\chi}$ represents a set of the robot poses. $J(\boldsymbol{\chi})$ denotes the objective function of the pose graphs in (2). The second term on the right side is the constraint on the relative pose of the robot at encounter nodes $\boldsymbol{x}_A$. $\boldsymbol{\delta}_A$ denotes the robot pose relative to the truck at the encounter nodes. The third term on the right side is the constraint on the truck pose (green arrow in Figure 4). $\boldsymbol{\delta}^*$ denotes the truck pose. $\boldsymbol{\Omega}^{loop}$ and $\boldsymbol{\Omega}^*$ denote the information matrices. As the truck pose is typically obtained accurately, $\boldsymbol{\Omega}^*$ is set to a large value.

IV. Loop Detection

The method of encounter node detection during map correction (Section III. B) is similar to the method of revisit node detection (SectionIII. A). Therefore, in this section, we

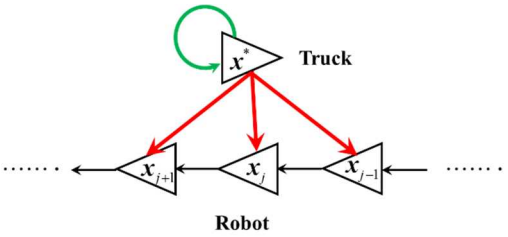


Figure 4. Pose graph for map correction.

describe the method of revisit node detection during local map building.

A. Detection of Candidate of Revisit Nodes

To detect revisit nodes, a candidate for revisit nodes is first obtained using the self-location information of the robot by NDT SLAM. If the distance of an old node from the current node is less than 10 m, the old node is recognized as a candidate for revisit nodes.

Thereafter, the Loop Probability Indicator (LPI) [18] is calculated using stationary scan data captured at the candidate for the revisit and current nodes. Each grid of the voxel map is first classified into three types: line, plane, or other voxels in Figure 5. Three eigenvalues ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$) are calculated from LiDAR scan data in voxels based on principal component analysis, and the following features are calculated:

$$q_1 = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}},\ q_2 = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}},\ q_3 = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{4}$$

When the maximum values are q_1, q_2, and q_3, the voxel is determined as being of line, plane, or other types.

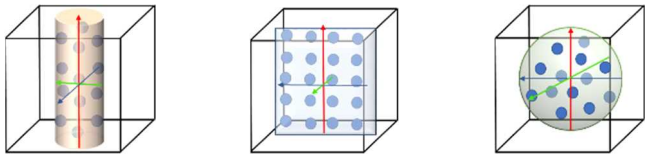
Based on the surface normal vector of the plane voxels, the plane voxels are further divided into nine classes: (1, 0, 0), (0, 1, 0), (0, 0, 1), $(1/\sqrt{2},1/\sqrt{2},0)$, $(1/\sqrt{2},-1/\sqrt{2},0)$, $(1/\sqrt{2},0,1/\sqrt{2})$, $(-1/\sqrt{2},0,1/\sqrt{2})$, $(0,1/\sqrt{2},1/\sqrt{2})$, and $(0,-1/\sqrt{2},1/\sqrt{2})$.

Two feature descriptors $\boldsymbol{U} = (u_1, u_2, \cdots, u_{11})^T$ and $\boldsymbol{V} = (v_1, v_2, \cdots, v_{11})^T$ are defined. $\boldsymbol{U}$ is calculated from LiDAR scan data captured at the candidate for revisit nodes, and $\boldsymbol{V}$ is calculated from the LiDAR scan data at the current node. u_1 and v_1 denote the numbers of line voxels in the voxel map. $u_2 - u_{10}$ and $v_2 - v_{10}$ denote the numbers of plane voxels divided into nine classes. u_{11} and v_{11} denote the numbers of other voxels.

From the feature descriptors $\boldsymbol{U}$ and $\boldsymbol{V}$, the LPI is given by

$$\mathrm{LPI} = \frac{\sum_{i=1}^{11} \{\max(u_i, v_i) - |u_i - v_i|\}}{\sum_{i=1}^{11} \max(u_i, v_i)} \tag{5}$$

A higher degree of similarity between the LiDAR scan data at both nodes leads to a larger LPI. Thus, the loop can be detected from the candidate of the revisit nodes using a large LPI value (a threshold of 80% in this study).



(a) Line voxel (b) Plane voxel (c) Other voxel

Figure 5. Classification of voxels.

B. Detection of Revisit Nodes and Calculation of Relative Pose

Revisit nodes are determined from the candidate for revisit nodes using a Matching Distance Indicator (MDI). From two LiDAR scan data captured at the current node and each candidate for revisit nodes, the relative pose of the robot is calculated using NDT scan matching. The MDI is then given:

$$\mathrm{MDI} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{6}$$

where N represents the number of measurements in the LiDAR scan data captured at the candidate for revisit nodes. d_i denotes the nearest neighbor distance.

A higher degree of similarity between the LiDAR scan data captured at two nodes leads to a smaller MDI. The loop can then be detected by a smaller MDI value (a threshold of 1.5 m in this study).

In NDT scan matching, if an initial value of the relative pose is given incorrectly, both the relative pose estimate and MDI become inaccurate due to local minima issues. To correctly set an initial value of the relative pose, an FPFH [16] is used.

Point features are extracted using FPFH from two LiDAR scan data captured at the current node and each candidate for revisit nodes. First, LiDAR scan data (stationary scan data) captured at the current node are mapped onto a voxel map (grid size of 0.2 m) in Σ_b and downsampled using a voxel grid filter. The centroid of the stationary scan data in the i-th voxel (i = 1, 2, …) on the voxel map is then obtained. The centroid is called the feature point A_i. From stationary scan data captured at each candidate for revisit nodes, the feature point B_i is obtained in the same way in Σ_b.

Point feature histograms (33 dimensions in this study) are calculated based on the feature points A_i and B_i, and their feature points are matched as follows:

Step 1: The three-feature point A_i (i = 1, 2, 3) is randomly extracted from the set of feature points obtained at the current scan. Then, 100 feature points B_j (j = 1, 2, …, 100) with similar feature histograms as those of A_i are extracted using the k-nearest neighbor method from the set of feature points obtained by each candidate for revisit nodes. We denote the triangle consisting of the three-feature point $\{A_1, A_2, A_3\}$ as A', while that consisting of any three-feature points from 100 feature points B_i as B'. The three-feature point $\{B_1, B_2, B_3\}$ is selected so that the two triangles A' and B' are congruent.

Step 2: The pose of the candidate for revisit node relative to the current node is denoted by $\delta\boldsymbol{X} = (\delta x, \delta y, \delta z, \delta\phi, \delta\theta, \delta\psi)^T$, where $\delta\boldsymbol{x} = (\delta x, \delta y, \delta z)^T$ and $\delta\boldsymbol{\theta} = (\delta\phi, \delta\theta, \delta\psi)^T$ denote the relative position and attitude angle (roll, pitch, and yaw angles), respectively.

In the matched triangles A' and B', the centroid positions of the three-feature points $\{A_1, A_2, A_3\}$ and $\{B_1, B_2, B_3\}$ are denoted by $\bar{\boldsymbol{a}}$ and $\bar{\boldsymbol{b}}$, respectively. The feature point matrices are denoted by $\delta\boldsymbol{a} = (\delta\boldsymbol{a}_1, \delta\boldsymbol{a}_2, \delta\boldsymbol{a}_3)^T$ and $\delta\boldsymbol{b} = (\delta\boldsymbol{b}_1, \delta\boldsymbol{b}_2, \delta\boldsymbol{b}_3)^T$, where $\delta\boldsymbol{a}_i \triangleq \boldsymbol{a}_i^* - \bar{\boldsymbol{a}}$ and $\delta\boldsymbol{b}_i \triangleq \boldsymbol{b}_i^* - \bar{\boldsymbol{b}}$,

 ISBN: 978-1-68558-089-6

and $\boldsymbol{a}_i^*$ and $\boldsymbol{b}_i^*$ are the 3D positions of the feature points A_i and B_i, respectively. Based on the matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, which are defined by the singular value decomposition ($\boldsymbol{H} = \boldsymbol{W}_1 \boldsymbol{\Sigma} \boldsymbol{W}_2^T$) of the matrix $\boldsymbol{H} = \delta\boldsymbol{b} \bullet \delta\boldsymbol{a}^T$, the relative position $\delta \boldsymbol{x}$ and the rotational matrix $\boldsymbol{R}(\delta\boldsymbol{\theta})$ related to the relative attitude angle $\delta\boldsymbol{\theta}$ are given by

$$\begin{aligned}\boldsymbol{R}(\delta\boldsymbol{\theta}) &= \boldsymbol{W}_2 \boldsymbol{W}_1^T \\ &= \begin{pmatrix} \cos\delta\theta\cos\delta\psi & \sin\delta\phi\sin\delta\theta\cos\delta\psi - \cos\delta\phi\sin\delta\psi & \cos\delta\phi\sin\delta\theta\cos\delta\psi + \sin\delta\phi\sin\delta\psi \\ \cos\delta\theta\sin\delta\psi & \sin\delta\phi\sin\delta\theta\sin\delta\psi + \cos\delta\phi\cos\delta\psi & \cos\delta\phi\sin\delta\theta\sin\delta\psi - \sin\delta\phi\cos\delta\psi \\ -\sin\delta\theta & \sin\delta\phi\cos\delta\theta & \cos\delta\phi\cos\delta\theta \end{pmatrix} \end{aligned} \quad (7)$$

$$\delta\boldsymbol{x} = \bar{\boldsymbol{a}} - \boldsymbol{R}(\delta\boldsymbol{\theta})\bar{\boldsymbol{b}} \quad (8)$$

Based on the relative pose, the 3D position $\boldsymbol{b}_i$ of the feature point B_i in Σ_w can be transformed to the 3D position $\boldsymbol{b}_i' = \boldsymbol{R}(\delta\boldsymbol{\theta})\boldsymbol{b}_i + \delta\boldsymbol{x}$ in Σ_b. The feature point nearest to $\boldsymbol{b}_i'$ is extracted from the set of feature points A_i (i =1, 2, ...), and the 3D position of the nearest feature point is denoted by $\tilde{\boldsymbol{a}}_i$. Then, the cost function is given by

$$J = \frac{1}{N_B}\sum_{i=1}^{N_B} (\tilde{\boldsymbol{a}}_i - \boldsymbol{b}_i')^T (\tilde{\boldsymbol{a}}_i - \boldsymbol{b}_i') \quad (9)$$

where N_B represents the number of the feature points B_i.

Step 3: Steps 1 and 2 are repeated 100 times to find the relative pose $\delta\boldsymbol{X}$ with the smallest J in (9). Then, the relative pose $\delta\boldsymbol{X}_0$ is obtained. In NDT scan matching, the relative pose $\delta\boldsymbol{X}_0$ is used as the initial value, and the iterative calculation is performed. Therefore, the accurate relative pose is calculated, and the MDI in (6) is accurately obtained.

V. FUNDAMENTAL EXPERIMENTS

Mapping experiments are conducted on our university campus, as depicted in Figure 6. A truck stops at the yellow circle in Figure 6, and the robot starts from the yellow circle, moves on the red and green paths in areas 1 and 2, and returns to the yellow circle. LiDAR and IMU data of the truck-and-robot system are recorded, and mapping is performed offline.

The distances travelled by the robot in areas 1 and 2 are 250 and 95 m. respectively, and the maximum velocity is approximately 5 km/h. Figure 7 depicts the attitude angle of the robot during movement, which is observed by the IMU.

For comparison, maps are built in the following cases:

Case 1: NDT SLAM-based local map building using robot-mounted LiDAR,

Case 2: NDT SLAM-based local map building without distortion correction of LiDAR scan data,

Case 3: NDT Graph SLAM-based local map building,

Case 4: Correction of local map using truck-mounted LiDAR.

Note that, in cases 1, 3, and 4, the distortion correction

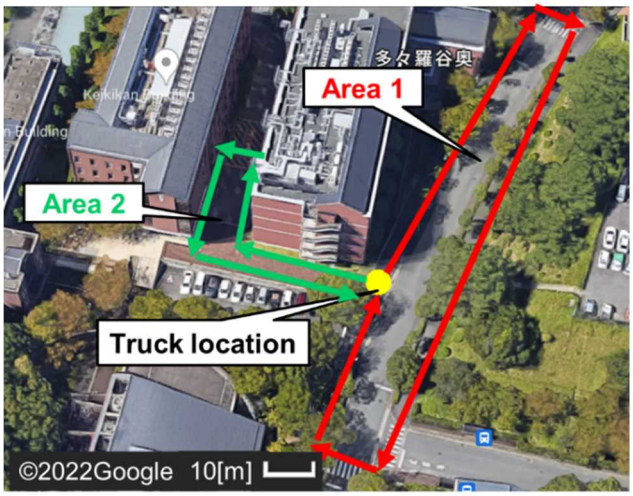
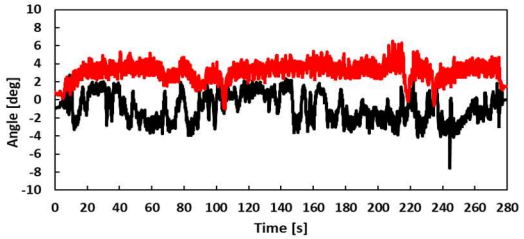
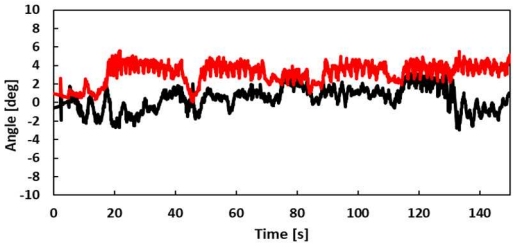


Figure 6. Experimental environment. The yellow circle indicates the truck location and start/goal position of robot. The red and green lines indicate the movement paths of robot.



(a) Area 1



(b) Area 2

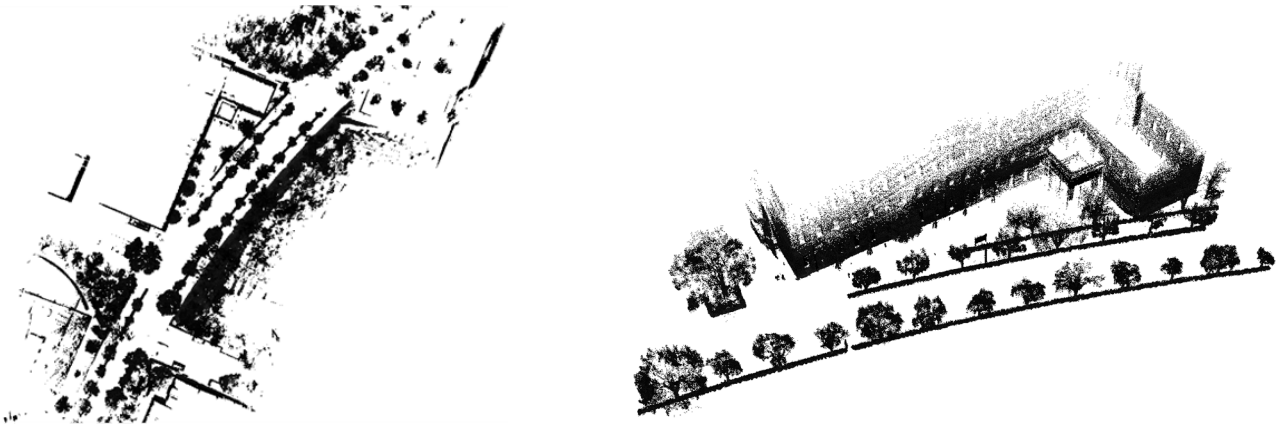
Figure 7. Roll (black) and pitch (red) angles of robot.

method is implemented.

Figures 8 and 9 show the mapping results in areas 1 and 2 (local maps 1 and 2), respectively, using case 4. These figures show that the proposed method can build an environmental map.

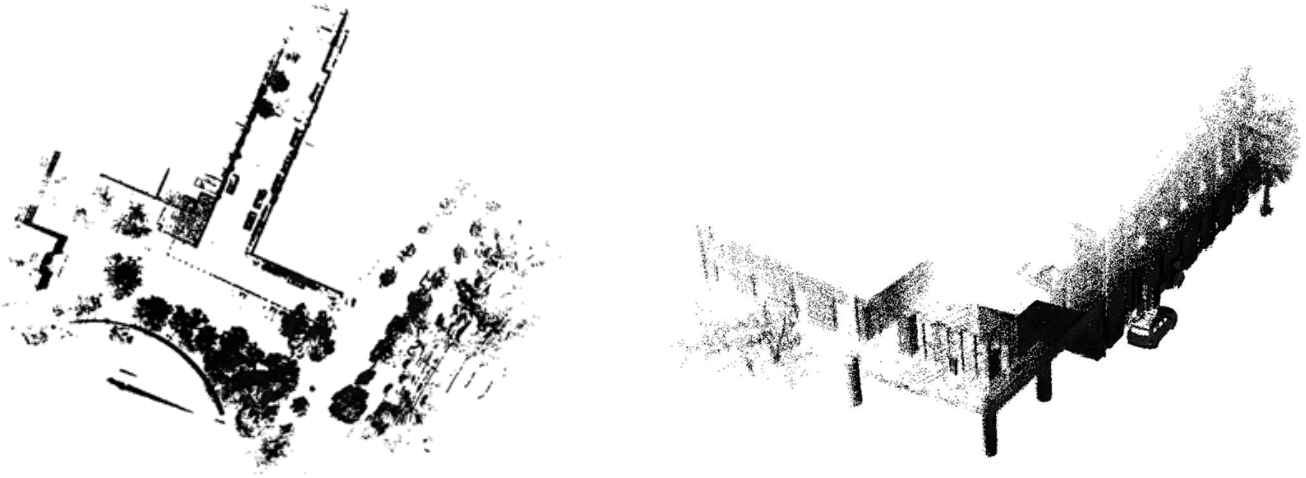
In SLAM-based mapping, the mapping accuracy is equivalent to that of the self-pose estimate of the robot. Therefore, to evaluate the mapping accuracy, the error of position estimate of the robot at the goal position is measured using a GNSS/LiDAR positioning system installed on the truck.

Tables I and II show the results in areas 1 and 2, respectively, shown in Figure 6, where the robot moves twice in each area. From theses tables, we can conclude that case 3



(a) Overall map (top view) (b) Enlarged map (bird's-eye view)

Figure 8. Mapping result in area 1 (local map 1).



(a) Overall map (top view) (b) Enlarged map (bird's-eye view)

Figure 9. Mapping result in area 2 (local map 2).

TABLE I. ERROR IN POSITION ESTIMATE OF ROBOT AT GOAL POSITION (LOCAL MAP 1).

	CASE 1	CASE 2	CASE 3	CASE 4
Run 1	3.09 m	3.63 m	0.15 m	0.13 m
Run 2	3.30 m	4.54 m	0.10 m	0.10 m

TABLE II. ERROR IN POSITION ESTIMATE OF ROBOT AT GOAL POSITION (LOCAL MAP 2).

	CASE 1	CASE 2	CASE 3	CASE 4
Run 1	0.92 m	1.17 m	0.28 m	0.10 m
Run 2	0.47 m	1.89 m	0.25 m	0.13 m

provides better results than cases 1 and 2. Furthermore, case 4 provides better results than case 3.

VI. CONCLUSION AND FUTURE WORK

This paper presented a LiDAR SLAM-based mapping method in truck-and-robot system for last-mile delivery systems. Distortion in scan data from robot-mounted LiDAR was corrected using a Kalman filter-based method. LiDAR scan data related to stationary objects were extracted from corrected scan data using an occupancy grid-based method, and local maps were built using NDT Graph SLAM.

Furthermore, a feature-based loop detection method was presented using surface features and FPFH. The local map was corrected in the Graph SLAM framework using scan data from truck-mounted LiDAR. The efficacy of the presented

 ISBN: 978-1-68558-089-6

mapping method was demonstrated through experimental results obtained in our university campus.

We are currently performing quantitative evaluations of the proposed method in various environments. In future works, map building using small and lightweight solid-state LiDAR instead of the mechanical LiDAR used in this paper will be performed. In addition, map update and maintenance will be studied.

REFERENCES

[1] E. Shaklab et al., "Towards Autonomous and Safe Last-mile Deliveries with AI-augmented Self-driving Delivery Robots," arXiv:2305.17705, 2023.

[2] J. Hooks et al., "ALPHRED: A Multi-Modal Operations Quadruped Robot for Package Delivery Applications," IEEE Robotics and Automation Letters, vol. 5, pp. 5409-5416, 2020.

[3] V. Balaska et al., "A Viewpoint on the Challenges and Solutions for Driverless Last-Mile Delivery," Machines, vol. 10, pp. 1–15, 2022.

[4] A. Heimfarth, M. Ostermeier, and A. Hübner, "A Mixed Truck and Robot Delivery Approach for the Daily Supply of Customers," European J. Operational Research, vol. 303, pp. 401–421, 2022.

[5] L. Qingqing, J. P. Queralta, T. N. Gia, Z. Zou, and T. Westerlund, "Multi Sensor Fusion for Navigation and Mapping in Autonomous Vehicles: Accurate Localization in Urban Environments," arXiv:2013.13719, 2021.

[6] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," IEEE Access, vol. 8, pp. 58443–58469, 2020.

[7] B. Huang, J. Zhao, and J. Liu, "A Survey of Simultaneous Localization and Mapping," eprint arXiv:1909.05214, 2019.

[8] S. Kuutti et al., "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," IEEE Internet of Things Journal, vol. 5, pp. 829–846, 2018.

[9] S. Tanaka, C. Koshiro, M. Yamaji, M. Hashimoto, and K. Takahashi, "Point Cloud Mapping and Merging in GNSS-Denied and Dynamic Environments Using Only Onboard Scanning LiDAR," Int. J. Advances in Systems and Measurements, vol. 13, pp. 275–288, 2020.

[10] K. Matsuo, A. Yoshida, M. Hashimoto, and K. Takahashi, "Normal Distributions Transform-based Mapping Using Scanning LiDAR Mounted on Motorcycle," Proc. of the 5th Int. Conf. on Advances in Sensors, Actuators, Metering and Sensing, pp. 69–75, 2020.

[11] I. Yoshida, A. Yoshida, M. Hashimoto, and K. Takahashi, "Map Building Using Helmet-Mounted LiDAR for Micro-Mobility," Artificial Life and Robotics, vol. 28, pp. 471–482, 2023.

[12] S. Hong, H. Ko, and J. Kim, "VICP: Velocity Updating Iterative Closest Point Algorithm," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1893–1898, 2010.

[13] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-LOAM: Truncated Least Squares LiDAR-only Odometry and Mapping in Real Time," IEEE Trans. on Geoscience and Remote Sensing, vol. 60, pp. 1–13, 2022.

[14] S. Arshad and G. W. Kim, "Role of Deep Learning in Loop Closure Detection for Visual and Lidar SLAM: A Survey," Sensors, vol. 21, p. 1–17, 2021.

[15] L. Huang, "Review on LiDAR-based SLAM Techniques," Proc. Int. Conf. on Signal Processing and Machine Learning, pp. 163–168, 2021.

[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," Proc. of IEEE/RSJ Int. Conf. on Robotics and Automation, pp. 3212–3217, 2009.

[17] P. Biber and W. Strasser, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2743–2748, 2003.

[18] F. Martín, R. Triebel, L. Moreno, and R. Siegwart, "Two Different Tools for Three-Dimensional Mapping: DE-based Scan Matching and Feature-Based Loop Detection," Robotica, vol. 32, pp. 19–41, 2017.

 ISBN: 978-1-68558-089-6