Point Cloud Mapping and Merging in GNSS-Denied and Dynamic Environments Using Onboard Scanning LiDAR

Seiya Tanaka, Chisato Koshiro, Misato Yamaji
Graduate School of Science and Engineering
Doshisha University
Kyotanabe, Kyoto 610-0394 Japan
e-mail: {ctwd0144, ctwf0118, ctwf0148}@mail4.doshisha.ac.jp

Masafumi Hashimoto, Kazuhiko Takahashi
Faculty of Science and Engineering
Doshisha University
Kyotanabe, Kyoto 610-0394 Japan
e-mail: {mhashimo, katakaha}@mail.doshisha.ac.jp

Abstract— **This paper presents a 3D point cloud mapping and merging in Global Navigation Satellite Systems (GNSS)-denied and dynamic environments using only a scanning Light Detection And Ranging (LiDAR) mounted on a vehicle. Distortion in scan data from the LiDAR is corrected by estimating the vehicle's pose (3D positions and attitude angles) in a period shorter than the LiDAR scan period using Normal Distributions Transform (NDT) scan matching and Extended Kalman Filter (EKF). The corrected scan data are mapped onto an elevation map. Static and moving scan data, which originate from static and moving objects, respectively, in the environments are classified using the occupancy grid method. Only the static scan data are utilized to generate several submaps in different small areas using NDT-based Simultaneous Localization And Mapping (NDT SLAM) and Graph SLAM. These submaps are merged using Graph SLAM. Experimental results obtained in outdoor residential and urban road environments show the LiDAR-based mapping and merging via EKF and NDT-Graph SLAM provide accurate maps in GNSS-denied and dynamic environments.**

Keywords-LiDAR; point cloud map; mapping and merging; NDT-Graph SLAM.

I. INTRODUCTION

This paper is an extended and improved version of an earlier paper presented at the IARIA Conference on Systems (ICONS 2020) [1] in Lisbon.

Recently, many studies have been conducted on the autonomous driving and active safety of vehicles, such as automobiles and personal mobility vehicles, and on autonomous robots for last-mile and first-mile automation. Important technologies from these studies include environmental map generation (mapping) [2] and map-matching-based self-pose estimation by vehicles using generated maps [3]. Many related studies used cameras, radars, and Light Detection And Ranging (LiDAR) [4][5].

In this paper, we focus on mapping with a scanning LiDAR mounted on a vehicle. Compared with camera-based mapping, LiDAR-based mapping is robust to lighting conditions and requires less computational time. Furthermore, the accuracy of LiDAR-based mapping is better than that of radar-based mapping due to the higher spatial resolution of LiDAR. For these reasons, we focus on LiDAR-based mapping.

In Intelligent Transportation Systems (ITS) domains, mobile mapping systems are used for mapping in wide road environments, such as highways and motorways [6]. We studied a method for point cloud mapping in narrow road environments, such as residential roads in urban and mountainous environments, using only a vehicle-mounted LiDAR [7]. The generated map could be applied to the autonomous driving and navigation of various smart vehicles, such as intelligent wheelchairs, personal mobility devices, and delivery robots [8]. The generated maps may also be utilized in various social services, such as disaster prevention and mitigation.

Although mapping systems often utilize position information from Global Navigation Satellite Systems (GNSS) [9], the accuracy of GNSS positioning is decreased in urban and mountainous areas due to the blockage, reflection, and diffraction caused by buildings and mountains. In addition, mapping systems designed for mapping in static environments generate inconsistent maps in practical dynamic environments that have moving objects, such as cars and pedestrians.

To address these problems, many studies have been conducted on LiDAR-based Simultaneous Localization And Mapping (SLAM) [9]. However, LiDAR-based SLAM in GNSS-denied and dynamic environments, such as urban street canyons in which the GNSS accuracy deteriorates and vehicles and people move, remains a significant challenge. This paper presents a point cloud mapping that uses only an onboard scanning LiDAR in GNSS-denied and dynamic environments. To do so, this technique integrates three methods that we previously proposed: distortion correction of the LiDAR scan data [10], extraction of scan data related to static objects from the entire LiDAR scan data [11][12], and point cloud mapping based on Normal Distributions Transform (NDT) and Graph-based SLAM [7]. The mapping performance by the proposed method is shown through experimental results in outdoor road environments.

The rest of this paper is organized as follows. Section II presents an overview of related work, and Section III describes the experimental system. Section IV explains the correction method of LiDAR scan data distortion, and Section V presents the extraction method of static scan data, which are related to static objects (removal of moving scan data, which are related to moving objects) from the entire LiDAR scan data. Section VI describes the mapping and merging methods based on NDT and Graph SLAM (called NDT-Graph SLAM). Finally, Section VII explains the experiments conducted to show the performance of our method, followed by the conclusions in Section VIII.

II. RELATED WORK

The main contribution of this paper is the conduct of LiDAR-based SLAM in GNSS-denied and dynamic environments by integrating components that we previously proposed: distortion correction of the LiDAR scan data [10], extraction of the scan data related to static objects from the entire LiDAR scan data [11][12], and point cloud mapping and merging based on NDT-Graph SLAM [7].

LiDAR-based SLAM is performed by mapping LiDAR scan data captured in a sensor coordinate frame onto a world coordinate frame using the vehicle's self-pose (position and attitude angle) information. The LiDAR obtains range measurements by scanning LiDAR beams. Thus, when the vehicle moves, the entire scan data within one scan (LiDAR beam rotation of 360° in a horizontal plane) cannot be obtained at the same pose of the vehicle. Therefore, if the entire scan data obtained within one scan are mapped onto the world coordinate frame using information about the vehicle's pose at a single point in time, distortion will arise in mapping. This distortion can be corrected by determining the vehicle's pose more frequently than the LiDAR scan period, i.e., for every LiDAR measurement in the scan.

Many distortion correction methods have been proposed [13][14][15]. However, most methods used additional sensors, such as odometer, Inertial Measurement Unit (IMU), and GNSS. Simple interpolation algorithms were also applied to determine a vehicle's pose more frequently than the LiDAR scan period. Unlike conventional methods, we corrected the distortion of LiDAR scan data using only the LiDAR information via Extended Kalman Filter (EKF) [10]. Our distortion correction method performed well.

When environmental features such as planes and pole-like objects are available, scan matching (such as NDT [16] and Iterative Closest Points (ICP) [17] methods) is applied to LiDAR-based SLAM in GNSS-denied environments [18]. Scan matching is adopted to calculate the transformation between LiDAR scans. The LiDAR-based SLAM is then performed based on the calculated continuous transformation. One of cons in the LiDAR-based SLAM is the drift (degradation of the accuracy over time) due to the accumulation error. To reduce the drift, Graph SLAM [19] is employed in conjunction with LiDAR-based SLAM. Another effective approach toward reducing the drift by LiDAR-based SLAM is submap generation and merging; the drift can be avoided by allowing short trajectories per submap [20][21].

We presented a mapping method in GNNS denied environments based on NDT-Graph SLAM [7]. A vehicle equipped with a LiDAR was moved such that loops could be made in road networks, and several submaps (maps of different small areas) were generated using NDT-Graph SLAM. Several submaps were also merged using Graph SLAM. Such approach to submap generation and merging makes it easy to update and maintain maps. However, further improvement is needed in the accuracy of submap merging. In addition, since a static world was assumed in our previous work, the presence of moving objects in practical dynamic environments deteriorates mapping performance. Then, improvements are required in the mapping method in dynamic environments.

In dynamic environments, LiDAR scan data can be classified into two types, namely, scan data originating from moving objects (moving scan data), and those originating from static objects (static scan data), such as buildings, trees, and traffic poles. For accurate mapping, the moving scan data have to be removed; only the static scan data will be utilized. This problem is addressed by SLAM-Moving Object Tracking (MOT) or SLAM-Detection And Tracking of Moving Objects (DATMO) approaches [22][23].

Apart from mapping, we have studied MOT and DATMO in crowded dynamic environments [11][12] for driving safety. Our moving-object detection method in MOT and DATMO was based on the occupancy grid method, which used the cell occupancy time and is simpler than usual probabilistic occupancy grid methods [24]. Our moving-object detection method will accurately remove moving scan data from the entire LiDAR scan data captured in dynamic environments and generate static maps.

III. EXPERIMENTAL SYSTEM

As shown in Figure 1, our small experimental vehicle is equipped with a 32-layer scanning LiDAR (Velodyne HDL-32E). 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 LiDAR provides 384 measurements (the object's 3D position and reflection intensity) every 0.55 ms (at 2° horizontal angle increments). The period for the LiDAR beam to complete one rotation (360°) in the horizontal direction is 100 ms, and 70,000 measurements are obtained in one rotation.

In this paper, one rotation of the LiDAR beam in the horizontal direction (360°) is considered one scan, and the data obtained from this scan are called scan data. Moreover, the LiDAR scan period (100 ms) is denoted as τ and the

Figure 1. Overview of experimental vehicle.

scan-data observation period (0.55 ms) as $\Delta\tau$.

To evaluate the SLAM performance, the vehicle is equipped with a GNSS/Inertial Navigation System (INS) unit (Novatel SPAN-CPT). The GNSS/INS unit outputs the vehicle's 3D position and attitude angle (roll, pitch, and yaw angles) every 100 ms. The horizontal and vertical position errors (Root Mean Square, RMS) are 0.02 m and 0.03 m, respectively. The roll and pitch angle errors (RMS) are both 0.02°, and the yaw angle error (RMS) is 0.06°.

IV. DISTORTION CORRECTION OF LIDAR SCAN DATA

This section describes the mapping method of scan data using NDT scan matching and distortion correction of LiDAR scan data using EKF.

A. NDT Scan Matching

The vehicle coordinate frame Σ_b (O_b-$x_b y_b z_b$) is defined in Figure 2. The origin O_b is the center of the rear wheel axle of the vehicle; the x_b, y_b, and z_b axes are the heading direction, the direction of the rear wheel axle, and the direction toward the sky, respectively. Although the LiDAR scan data are captured by the sensor coordinate frame fixed at the LiDAR, the objects' 3D positions in the scan data are always transformed to those in Σ_b . For convenience, the scan data are hereafter assumed to be captured in Σ_b .

When LiDAR scan data are captured in one scan, the scan data related to road surfaces are first removed using a method described in Section V, and the scan data related to objects are mapped onto a 3D grid map (voxel map) represented in Σ_b . A voxel grid filter [25] 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.

A local coordinate frame Σ_W (O_W -$x_W y_W z_W$) is defined in Figure 2. Σ_W coincides with Σ_b when the vehicle starts to generate the submap. In Σ_w , a voxel map with a voxel size of 1 m is used for NDT scan matching. 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 then given:

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

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

$$T(\boldsymbol{x}) = \begin{pmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & x \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & y \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The scan data obtained at the current time $t\tau$ (t = 0, 1, 2, …), $\boldsymbol{p}_{b(t)} = \{\boldsymbol{p}_{b1(t)}, \boldsymbol{p}_{b2(t)}, \cdots\}$, are called the new input scan, and the scan data obtained in the previous time, i.e., before $(t-1)\tau$, $\boldsymbol{P} = \{\boldsymbol{P}_{(0)}, \boldsymbol{P}_{(1)}, \cdots, \boldsymbol{P}_{(t-1)}\}$, are called the reference scan (environmental map).

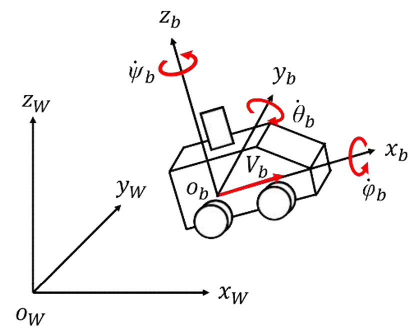


Figure 2. Notation related to vehicle motion.

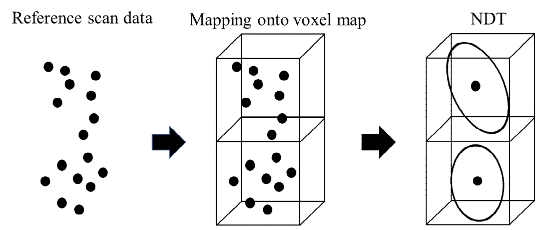


Figure 3. Normal distributions transform of reference scan data.

NDT scan matching [16] conducts a normal distribution transformation for the reference scan data in each grid on a voxel map. It calculates the mean and covariance of the LiDAR measurement positions, as shown in Figure 3. The vehicle's pose $\boldsymbol{x}_{(t)}$ at $t\tau$ is determined by matching the new input scan at $t\tau$ with the reference scan data obtained prior to $(t-1)\tau$. The vehicle's pose can be calculated by maximizing the following likelihood function:

$$\Lambda = \prod_{i=1}^{n} \exp\left(-\frac{1}{2} (\boldsymbol{p}_i(t) - \boldsymbol{q}_i)^T \boldsymbol{\Omega}_i^{-1} (\boldsymbol{p}_i(t) - \boldsymbol{q}_i) \right) \qquad (2)$$

where $\boldsymbol{q}_i$ and $\boldsymbol{\Omega}_i$ are the mean and covariance, respectively, of the reference scan in the i-th voxel. $\boldsymbol{p}_i$ is the new input scan in the i-th voxel.

The vehicle's pose is used for conducting a coordinate transform with (1). The new input scan can then be mapped to Σ_W , and the reference scan is updated. The downsized scan data are only used to calculate the vehicle's pose using NDT scan matching for a small computational cost.

In this study, we use the Point Cloud Library (PCL) [26] for NDT scan matching.

B. Distortion Correction of LiDAR Scan Data

A motion model of the vehicle is first described for the EKF-based correction of LiDAR scan data distortion.

As shown in Figure 2, the vehicle's linear velocity in Σ_b is defined as V_b (the velocity in the x_b-axis direction), and the angular velocities about the x_b, y_b, and z_b axes are defined as $\dot{\phi}_b$, $\dot{\theta}_b$, and $\dot{\psi}_b$, respectively. If the vehicle is assumed to move at nearly constant linear and angular velocities, the following motion model can then be derived:

$$\begin{pmatrix} x_{(t+1)} \\ y_{(t+1)} \\ z_{(t+1)} \\ \phi_{(t+1)} \\ \theta_{(t+1)} \\ \psi_{(t+1)} \\ V_{b(t+1)} \\ \dot{\phi}_{b(t+1)} \\ \dot{\theta}_{b(t+1)} \\ \dot{\psi}_{b(t+1)} \end{pmatrix} = \begin{pmatrix} x_{(t)} + a_{1(t)}\cos\theta_{(t)}\cos\psi_{(t)} \\ y_{(t)} + a_{1(t)}\cos\theta_{(t)}\sin\psi_{(t)} \\ z_{(t)} - a_{1(t)}\sin\theta_{(t)} \\ \phi(t) + a_2(t) + \{a_3(t)\sin\phi(t) + a_4(t)\cos\phi(t)\}\tan\theta(t) \\ \theta(t) + \{a_3(t)\cos\phi(t) - a_4(t)\sin\phi(t)\} \\ \psi(t) + \{a_3(t)\sin\phi(t) + a_4(t)\cos\phi(t)\}\dfrac{1}{\cos\theta(t)} \\ V_{b(t)} + \tau w_{\dot{V}_b} \\ \dot{\phi}_{b(t)} + \tau w_{\ddot{\phi}_b} \\ \dot{\theta}_{b(t)} + \tau w_{\ddot{\theta}_b} \\ \dot{\psi}_{b(t)} + \tau w_{\ddot{\psi}_b} \end{pmatrix} \quad (3)$$

where $a_1 = V_b\tau + \tau^2 w_{\dot{V}_b}/2$, $a_2 = \dot{\phi}_b\tau + \tau^2 w_{\ddot{\phi}_b}/2$, $a_3 = \dot{\theta}_b\tau + \tau^2 w_{\ddot{\theta}_b}/2$, and $a_4 = \dot{\psi}_b\tau + \tau^2 w_{\ddot{\psi}_b}/2$. $w_{\dot{V}_b}$, $w_{\ddot{\phi}_b}$, $w_{\ddot{\theta}_b}$, and $w_{\ddot{\psi}_b}$ are the acceleration disturbances.

Equation (3) is expressed in vector form as follows:

$$\boldsymbol{\xi}_{(t+1)} = \boldsymbol{f}[\boldsymbol{\xi}_{(t)}, \boldsymbol{w}, \tau] \quad (4)$$

where $\boldsymbol{\xi} = (x, y, z, \phi, \theta, \psi, V_b, \dot{\phi}_b, \dot{\theta}_b, \dot{\psi}_b)^T$ and $\boldsymbol{w} = (w_{\dot{V}_b}, w_{\ddot{\phi}_b}, w_{\ddot{\theta}_b}, w_{\ddot{\psi}_b})^T$.

The vehicle's pose obtained at $t\tau$ using NDT scan matching is defined as $\boldsymbol{z}_{NDT(t)} (= \hat{\boldsymbol{x}}_{(t)})$. The measurement equation is then

$$\boldsymbol{z}_{NDT(t)} = \boldsymbol{H}\boldsymbol{\xi}_{(t)} + \Delta\boldsymbol{z}_{NDT(t)} \quad (5)$$

where $\Delta\boldsymbol{z}_{NDT}$ is the measurement noise, and $\boldsymbol{H}$ is the following measurement matrix:

$$\boldsymbol{H} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \end{pmatrix}$$

The correction flow of LiDAR scan data is shown in Figure 4. The LiDAR scan period τ is 100 ms, and the scan-data observation period $\Delta\tau$ is 0.55 ms. When the scan data are mapped onto Σ_W using the vehicle's pose, which is calculated every LiDAR scan period, distortion arises in the environmental map. This distortion of the LiDAR scan data is therefore corrected by estimating the vehicle's pose using the EKF every scan-data observation period $\Delta\tau$.

The state estimate and its error covariance obtained at $(t-1)\tau$ using the EKF are denoted as $\hat{\boldsymbol{\xi}}_{(t-1)}$ and $\boldsymbol{\Gamma}_{(t-1)}$, respectively. From these quantities, the EKF gives the state prediction $\hat{\boldsymbol{\xi}}_{(t-1,1)}$ and its error covariance $\boldsymbol{\Gamma}_{(t-1,1)}$ at $(t-1)\tau + \Delta\tau$ as follows:

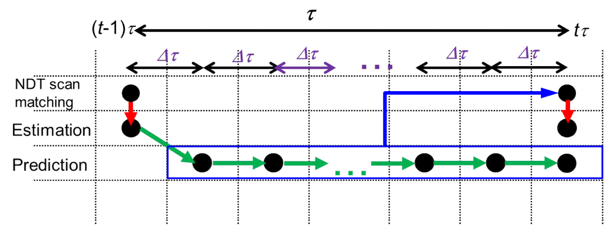


Figure 4. Flow of distortion correction.

$$\left.\begin{aligned} \hat{\boldsymbol{\xi}}_{(t-1,1)} &= \boldsymbol{f}[\hat{\boldsymbol{\xi}}_{(t-1)}, 0, \Delta\tau] \\ \boldsymbol{\Gamma}_{(t-1,1)} &= \boldsymbol{F}_{(t-1)}\boldsymbol{\Gamma}_{(t-1)}\boldsymbol{F}_{(t-1)}^T + \boldsymbol{G}_{(t-1)}\boldsymbol{Q}\boldsymbol{G}_{(t-1)}^T \end{aligned}\right\} \quad (6)$$

where $\boldsymbol{F} = \partial\boldsymbol{f}/\partial\hat{\boldsymbol{\xi}}$ and $\boldsymbol{G} = \partial\boldsymbol{f}/\partial\boldsymbol{w}$. $\boldsymbol{Q}$ is the covariance matrix of the plant noise $\boldsymbol{w}$.

By a similar calculation, the state prediction $\hat{\boldsymbol{\xi}}_{(t-1,j)}$ and its error covariance $\boldsymbol{\Gamma}_{(t-1,j)}$ at $(t-1)\tau + j\Delta\tau$ (where j = 1, 2, …,180) can be obtained by

$$\left.\begin{aligned} \hat{\boldsymbol{\xi}}_{(t-1,j)} &= \boldsymbol{f}[\hat{\boldsymbol{\xi}}_{(t-1,j-1}, 0, \Delta\tau] \\ \boldsymbol{\Gamma}_{(t-1,j)} &= \boldsymbol{F}_{(t-1,j-1)}\boldsymbol{\Gamma}_{(t-1,j-1)}\boldsymbol{F}_{(t-1,j-1)}^T \\ &\quad + \boldsymbol{G}_{(t-1,j-1)}\boldsymbol{Q}\boldsymbol{G}_{(t-1,j-1)}^T \end{aligned}\right\} \quad (7)$$

In the state prediction $\hat{\boldsymbol{\xi}}_{(t-1,j)}$, the elements related to the vehicle's pose $(x, y, z, \phi, \theta, \psi)$ are denoted as $\hat{\boldsymbol{X}}_{(t-1,j)}$. Using (1) and the pose prediction, the scan data $\boldsymbol{p}_{bi(t-1,j)}$ in Σ_b obtained at $(t-1)\tau + j\Delta\tau$ can be transformed to $\boldsymbol{p}_{i(t-1,j)}$ in Σ_W as follows:

$$\begin{pmatrix} \boldsymbol{p}_{i(t-1,j)} \\ 1 \end{pmatrix} = \boldsymbol{T}(\hat{\boldsymbol{X}}_{(t-1,j)}) \begin{pmatrix} \boldsymbol{p}_{bi(t-1,j)} \\ 1 \end{pmatrix} \quad (8)$$

Since the LiDAR scan period τ is 100 ms, and the scan-data observation period $\Delta\tau$ is 0.55 ms, the time $t\tau$ is equal to $(t-1)\tau + 180\Delta\tau$. Using the pose prediction $\hat{\boldsymbol{X}}_{(t-1,180)}$ at $t\tau$, the scan data $\boldsymbol{p}_{i(t-1,j)}$ at $(t-1)\tau + j\Delta\tau$ in Σ_W are transformed into the scan data $\boldsymbol{p}^*_{bi(t)}$ at $t\tau$ in Σ_b as follows:

$$\begin{pmatrix} \boldsymbol{p}^*_{bi(t)} \\ 1 \end{pmatrix} = T(\hat{\boldsymbol{X}}_{(t-1,180)})^{-1} \begin{pmatrix} \boldsymbol{p}_{i(t-1,j)} \\ 1 \end{pmatrix} \quad (9)$$

Using the corrected scan data $\boldsymbol{p}^*_{b(t)} = \{\boldsymbol{p}^*_{b1(t)}, \boldsymbol{p}^*_{b2(t)}, \cdots\}$ within one scan (LiDAR beam rotation of 360° in a horizontal plane) as the new input scan, NDT scan matching can accurately calculate the vehicle's pose $z_{NDT(t)}$ at $t\tau$. Based on (4) and (5), the EKF then gives the state estimate $\hat{\xi}_{(t)}$ and its error covariance $\boldsymbol{\Gamma}_{(t)}$ at $t\tau$ by

$$\left.\begin{aligned}\hat{\boldsymbol{\xi}}_{(t)} &= \hat{\boldsymbol{\xi}}_{(t-1,180)} + \boldsymbol{K}_{(t)}\{\boldsymbol{z}_{NDT(t)} - \boldsymbol{H}\hat{\boldsymbol{\xi}}_{(t-1,180)}\} \\ \boldsymbol{\Gamma}_{(t)} &= \boldsymbol{\Gamma}_{(t-1,180)} - \boldsymbol{K}_{(t)}\boldsymbol{H}\boldsymbol{\Gamma}_{(t-1,180)}\end{aligned}\right\} \quad (10)$$

where $\hat{\boldsymbol{\xi}}_{(t-1,180)}$ and $\boldsymbol{\Gamma}_{(t-1,180)}$ are the state prediction and its error covariance at $t\tau$ $(=(t-1)\tau+180\Delta\tau)$ respectively. $\boldsymbol{K}_{(t)} = \boldsymbol{\Gamma}_{(t-1,180)}\boldsymbol{H}^T \boldsymbol{S}^{-1}_{(t)}$ and $\boldsymbol{S}_{(t)} = \boldsymbol{H}\boldsymbol{\Gamma}_{(t-1,180)}\boldsymbol{H}^T + \boldsymbol{R}$. $\boldsymbol{R}$ is the covariance matrix of the measurement noise $\Delta\boldsymbol{z}_{NDT}$.

The corrected scan data $\boldsymbol{P}^*_{b(t)}$ are mapped onto Σ_W using the pose estimate calculated by (10), and the distortion in the environmental map can then be removed.

V. Extraction of Static Scan Data

In dynamic environments, which have moving objects, such as cars, two-wheelers, and pedestrians, LiDAR scan data related to moving objects (moving scan data) have to be removed from the entire scan data, and only scan data related to static objects (static scan data), such as buildings and trees, have to be utilized in mapping.

In the extraction of static scan data, the LiDAR scan data are classified into two types, namely, scan data originating from road surfaces (road-surface scan data) and those originating from objects (object scan data), based on the following rule-based method.

As shown in Figure 5, 32 measurements captured every horizontal resolution (0.16°) of the LiDAR are considered. The measurement r_1, which is the closest measurement to the LiDAR, is assumed to be the measurement belonging to road surfaces. We obtain the angle of a line connecting the adjacent measurements r_1 and r_2 relative to the *xy*-plane in Σ_W. If the angle is less than 15°, the measurement r_2 is determined to belong to road surfaces. If it is larger than 15°, the measurement r_2 is determined to belong to objects. By repeating this process for all LiDAR scan data, we can distinguish the scan data related to objects (blue points in Figure 5) and those related to road surfaces (red points). If the threshold for discriminating the scan data related to road surfaces and objects is small, slopes is mis-detected as objects. In general, the steep slope of vehicles is less than about 6°. The threshold is therefore set to 15°.

The object scan data are mapped onto an elevation map represented in Σ_W. In this paper, the cell of the elevation map is a square with a side length of 0.3 m. The height of each cell is the maximum height of multiple scan data mapped onto the cell.

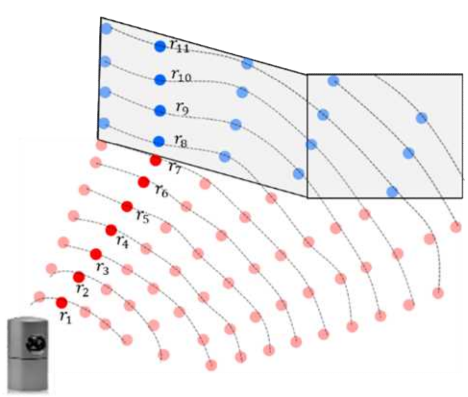


Figure 5. Extraction of LiDAR scan data related to objects.

A cell containing scan data is called an occupied cell. For the moving scan data, the time to occupy the same cell is short (less than 0.8 s in this paper), whereas for the static scan data, the time is long (not less than 0.8 s). Therefore, using the occupancy grid method, which is based on the cell occupancy time [11][12], the occupied cells are classified into two types of cells, namely, moving and static cells, which are occupied by the moving and static scan data, respectively. Cells that the LiDAR cannot identify because of obstructions are defined as unknown cells, and their cell occupancy time is not counted.

Since the scan data related to an object usually occupy multiple cells, adjacent occupied cells with almost the same height are clustered. In general, moving and static cells coexist in the same clustered cells. If the number of moving cells in clustered cells is not less than a threshold *TH*, these clustered cells are then decided as the moving-cell group; otherwise as the static-cell group. *TH* is given by the following sigmoid function:

$$TH = 0.5 + \frac{0.2}{1+\exp(5-0.3s)} \quad (11)$$

where s is the number of cells that constitute the cell group.

The above equation means that the threshold is dynamically determined to be 50 %–70 % according to the number of cells s. In our experience, since the speed of small (large) moving objects, such as pedestrians (cars), is low (high), the number of moving cells belonging to a cell group is small (large). To improve the performance of the moving-object detection, the threshold is set to 50 % (70 %) for small (large) objects with a small (large) number of occupied cells. The scan data in clustered static cells are applied to mapping.

When moving objects pause, such as vehicles pausing at red lights, the occupancy grid-based method often misidentifies their scan data as static scan data. To address this problem, road-surface scan data are mapped onto the elevation map, and the cells where the road-surface scan data have been occupied for several scans are determined as road-surface cells. If the road-surface cells contain object scan data, these data are always determined as moving scan data and removed from the entire scan data.

VI. Submap Generation and Merging

This section describes the methods of submap generation and merging based on NDT-Graph SLAM. For a clear explanation, we consider the generation and merging submaps 1 and 2, which are shown in Figure 6.

A. Submap Generation

In each submap, a local coordinate frame Σ_{Wi} (O_{Wi} -x_{Wi} y_{Wi} z_{Wi}) is defined, where i = 1, 2; Σ_{Wi} coincides with Σ_b when the vehicle starts to generate the submap i.

The vehicle's poses are mapped onto a factor graph (pose graph), as shown in Figure 7. In this figure, the vehicle's poses are represented as the graph nodes (black triangles), and the relative poses between two neighboring nodes are represented as the graph edges (black arrows). The vehicle's poses are calculated by NDT SLAM every 100 ms (LiDAR scan period).

To recognize whether or not the vehicle has already visited a place (called revisit node or loop), the candidate of the revisit nodes is first obtained using the self-location information of the vehicle, which is estimated by NDT SLAM. If the distance of an old node from the current node is smaller than 10 m, as shown in Figure 8, the old node is recognized as a candidate of the revisit nodes.

Thereafter, the Loop Probability Indicator (LPI) [27] is calculated using LiDAR scan data captured at the candidate of the revisit and current nodes. Each grid of the voxel map is first classified into three types of voxels: line, plane, and the other voxels in Figure 9. Three eigenvalues ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$) are calculated from LiDAR scan data in voxels based on the principal component analysis. When λ_2 / λ_1 is no more than 0.1, the voxel is decided as being of line type (Figure 9 (a)); when λ_3 / λ_2 is no more than 0.1, the voxel is decided as being of plane type (Figure 9 (b)); when λ_2 / λ_1 and λ_3 / λ_2 are more than 0.1, the voxel is decided as being of another type (Figure 9 (c)).

Based on the surface normal vector of the plane voxels in Σ_b, these 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 of the revisit nodes, and $\boldsymbol{V}$ is calculated from the LiDAR scan data at the current node. u_1 and v_1 are the numbers of line voxels in the voxel map. $u_2 - u_{10}$ and $v_2 - v_{10}$ are the numbers of plane voxels that are divided into nine classes. u_{11} and v_{11} are the numbers of the other voxels.

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

$$\mathrm{LPI} = \frac{\sum_{i=1}^{11} \{\max(u_i, v_i) - |u_i - v_i|\}}{\sum_{i=1}^{11} \max(u_i, v_i)} \tag{12}$$

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

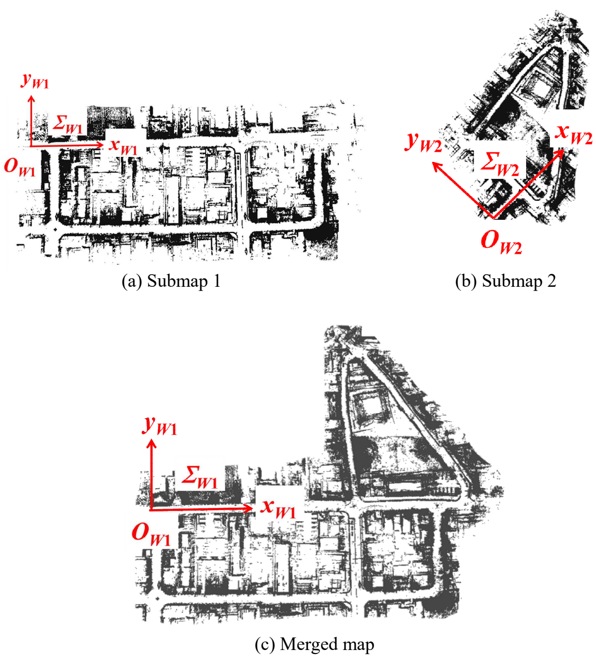


Figure 6. Submap generation and merging (top view).

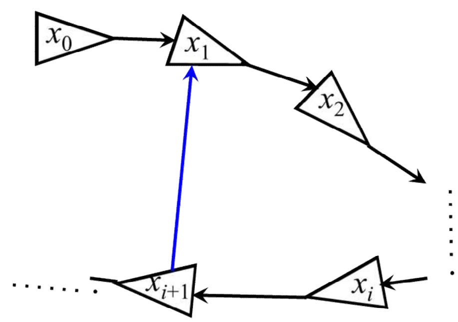


Figure 7. Pose graph in submap generation.

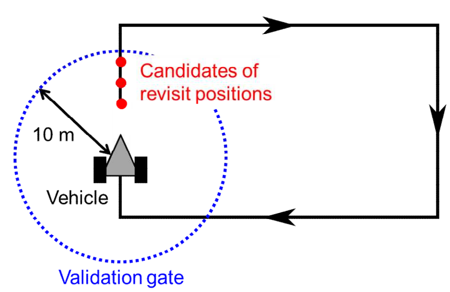


Figure 8. Loop closure detection in submap generation.

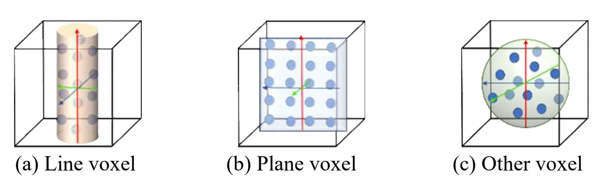


Figure 9. Classification of voxel.

The detection performance is then improved using a Matching Distance Indicator (MDI). From two LiDAR scan data captured at the current node and each candidate of the revisit nodes, the relative vehicle's pose is calculated based on NDT scan matching; the displacement of the self-locations at two nodes obtained by NDT SLAM is used as the initial relative pose for NDT scan matching.

In our experience, even if the relative pose of the vehicles at two nodes is large, a larger voxel size leads to a more robust matching in NDT scan matching. Therefore, the relative pose is calculated using two different voxel sizes. The relative pose is first calculated using a voxel size of 3 m. The obtained relative pose is used as the initial pose to calculate the relative pose by NDT scan matching with a voxel size of 1 m. The final estimate of the relative pose is applied to calculate the nearest neighbor distance between the two LiDAR scan data via NDT scan matching. The MDI is then calculated as

$$\mathrm{MDI} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{13}$$

where N is the number of measurements in the LiDAR scan data captured at the candidate of the revisit nodes. d_i is 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 closure can then be detected by a smaller MDI value (a threshold of 1.5 m in this paper).

When the loop closures are detected by both LPI and MDI, the current vehicle's pose relative to its pose at the revisit node is inputted to the pose graph as a loop closure constraint (blue arrow in Figure 7). The objective function of (14) is then minimized to improve the accuracy of submap generated by NDT SLAM:

$$\begin{aligned} J(\boldsymbol{\chi}) = &\sum_{i} \{(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) - \boldsymbol{\delta}_{i+1,i}\}^T \boldsymbol{\Omega}^{pose} \{(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) - \boldsymbol{\delta}_{i+1,i}\} \\ &+ \sum_{\boldsymbol{x}_A, \boldsymbol{x}_B \in \mathrm{loop}} \{(\boldsymbol{x}_B - \boldsymbol{x}_A) - \boldsymbol{\delta}_{A,B}\}^T \boldsymbol{\Omega}^{loop} \{(\boldsymbol{x}_B - \boldsymbol{x}_A) - \boldsymbol{\delta}_{A,B}\} \end{aligned} \tag{14}$$

where the first and second terms on the right side indicate the constraints on NDT SLAM and loop closure, respectively. $\boldsymbol{\chi} = (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_i^T, \cdots)^T$. $\boldsymbol{x}_i$ is the vehicle's pose at time $i\tau$. $\boldsymbol{\delta}_{i+1,i}$ is the relative pose of the vehicle between $i\tau$ and $(i+1)\tau$, which is calculated from NDT SLAM. $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ are the vehicle's poses at the revisit and current nodes, respectively. $\boldsymbol{\delta}_{A,B}$ indicates the relative pose of the vehicle at the two nodes, which is calculated from the LiDAR scan data using NDT scan matching. $\boldsymbol{\Omega}^{pose}$ and $\boldsymbol{\Omega}^{loop}$ are the information matrices; they are inverse covariance matrices of NDT SLAM and given based on [28].

In this paper, we apply the open-source software g2o [29] to generate pose graphs and optimize (14).

B. Submap Merging

We consider the merging of submaps 1 and 2 in Figure 6. Submap merging is performed by the following steps:

- Loop closure detection: detection of encounter nodes in pose graphs corresponding to the two submaps;
- Relative pose estimation: estimation of the relative pose of the two submaps using the LiDAR scan data at nodes encountered in the two pose graphs;
- Alignment: coordinate transform of submap 2 using the relative pose estimate to represent the submap in Σ_{W1}; and
- Merging: merging of the two submaps using pose graph optimization.

If there are three or more submaps, an enlarged submap is first made by merging the two submaps, and another submap is then merged with the enlarged submap. By repeating such process, three or more submaps can be merged.

The loop closures between submaps (intersession loop closures) are detected based on LPI and MDI. However, unlike the loop closure detection in each submap (intrasession loop closure), the self-location information of the vehicle estimated by NDT SLAM is not useless in narrowing down the candidate of the encounter nodes in the two pose graphs because two submaps are generated in different coordinate frames. It is thus assumed that all nodes in the two pose graphs are the candidate of the encounter nodes, and the LPI is calculated by the brute force method to narrow down the candidate of the encounter nodes. Therefore, if the numbers of nodes are N_1 and N_2 in the pose graphs corresponding to submaps 1 and 2, respectively, the LPI is calculated $N_1 \times N_2$ times.

As shown in Figure 10, we consider that two nodes (the i-th node in pose graph 1, which corresponds to submap 1, and the j-th node in pose graph 2, which corresponds to submap 2) are detected as the candidate of the encounter nodes by the LPI (a threshold of 80%). To determine using the MDI whether or not the candidate is that of the encounter nodes, the relative pose of the vehicle is calculated from two scan data in both nodes via NDT scan matching. However, since two submaps are generated in different local coordinate frames, the initial pose, which is used to accurately calculate the relative pose by NDT scan matching, is unknown.

At the j-th node in pose graph 2, the vehicle coordinate frame Σ_b, in which the LiDAR scan data are captured, is

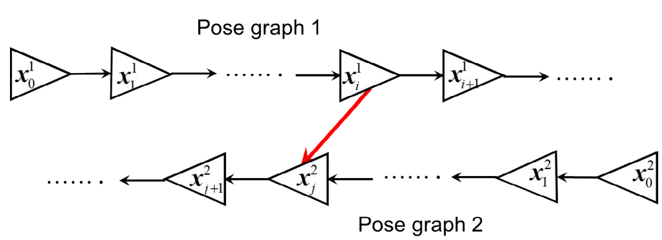


Figure 10. Pose graph in submap merging.

rotated about the z_b axis (yaw angle direction) in steps of 10° from 0° to 360° to address the above-mentioned problem. From the scan data at the i-th node in pose graph 1 and each of the 35 scan data at the j-th node in pose graph 2, NDT scan matching with an initial pose of zero value is applied to calculate the relative pose. If the relative pose estimate is correct, the MDI value will be small. The MDI for each of the 35 relative poses is then calculated, and the minimum MDI is selected. If the minimum MDI is 1.5 m or less, the candidate of the encounter nodes, the i-th and j-th nodes, is recognized as encounter nodes, and the relative pose is determined.

Such detection of intersession loop closure and related relative-pose calculation are repeated for all nodes in pose graphs 1 and 2. When many encounter nodes are detected in their pose graphs, the relative pose of the two pose graphs is determined by the weighted average of many relative poses. Using the relative pose, the coordinate transform of submap 2 is performed; consequently submaps 1 and 2 could be represented in the coordinate frame Σ_{W1}.

Finally, the relative pose of the vehicle at the encounter nodes is inputted to the pose graphs as the loop closure constraint (red arrow in Figure 10). The following objective function is then minimized to merge the two submaps:

$$J(\boldsymbol{\chi}^{total}) = J(\boldsymbol{\chi}^1) + J(\boldsymbol{\chi}^2) + \sum_{x_1, x_2 \in \text{loop}} \{(\boldsymbol{x}_2 - \boldsymbol{x}_1) - \boldsymbol{\delta}_{1,2}\}^T \boldsymbol{\Omega}_{1,2}^{loop} \{(\boldsymbol{x}_2 - \boldsymbol{x}_1) - \boldsymbol{\delta}_{1,2}\} \tag{15}$$

where $\boldsymbol{\chi}^{total} = (\boldsymbol{\chi}^{1T}, \boldsymbol{\chi}^{2T})^T$. $\boldsymbol{\chi}^1 = (\boldsymbol{x}_1^{1T}, \boldsymbol{x}_2^{1T}, \cdots, \boldsymbol{x}_i^{1T}, \cdots)^T$ and $\boldsymbol{\chi}^2 = (\boldsymbol{x}_1^{2T}, \boldsymbol{x}_2^{2T}, \cdots, \boldsymbol{x}_i^{2T}, \cdots)^T$ are sets of the vehicle's poses in pose graphs 1 and 2, respectively. $\boldsymbol{x}_i^1$ and $\boldsymbol{x}_j^2$ are the vehicle's poses at times $i\tau$ and $j\tau$, respectively. $J(\boldsymbol{\chi}^1)$ and $J(\boldsymbol{\chi}^2)$ are the objective functions of the pose graphs corresponding to submaps 1 and 2, respectively. The third term on the right side is the constraint on the vehicle's relative pose in the merging of the two pose graphs. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are encounter nodes in pose graphs 1 and 2, respectively. $\boldsymbol{\delta}_{1,2}$ indicates the relative pose of the vehicle at the encounter nodes, which is calculated from the LiDAR scan data captured at the nodes using NDT scan matching. $\boldsymbol{\Omega}_{1,2}^{loop}$ is the information matrix; it is inverse covariance matrix of NDT scan matching and given based on [28].

VII. Experimental Results

The performance of two methods is first examined, namely, distortion correction of LiDAR scan data and extraction of static scan data from the entire LiDAR scan data, which are presented in Sections IV and V, respectively. Thereafter, the mapping performance is shown through experimental results in residential and urban environments.

A. Performance of Distortion Correction of LiDAR Scan Data and Extraction of Static Scan Data

The experimental vehicle moves at a speed of about 40 km/h in two areas, as shown in Figures 11 (a) and (b). For comparison, the LiDAR scan data are mapped using NDT SLAM in the following cases:

Case 1: Mapping through the distortion correction of the LiDAR scan data and extraction of the static scan data from the entire scan data;

Case 2: Mapping without using either method.

Figures 12 and 13 show the mapping results on a straight road and an intersection area, respectively. The red line in (a) indicates the movement path of the experimental vehicle. The black and red dots in (b) and (c) indicate the static and moving scan data, respectively. These figures indicate that the extraction method of static scan data more significantly removes the tracks of cars. In the intersection, several cars slow down and stop at a red light or pause when turning left; they are determined as static objects. Consequently, in Figure 13 (b), LiDAR scan data related to cars partially remain.

Figure 14 shows the mapping result of a traffic sign in the road environment shown in Figure 12 (a). Figures 12–14 show that the mapping result obtained using the distortion correction of the LiDAR scan data is crisper than that obtained without using the distortion correction.

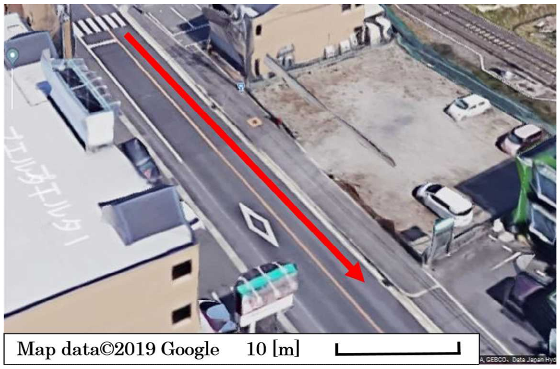
B. Mapping Performance

A mapping experiment is conducted in a residential environment near our university campus. The experimental vehicle moves at a speed of 10–20 km/h on a narrow road (6 m width) in the residential environment shown in Figure 15, and sensor data are recorded. The traveled distance of the vehicle is 2000 m. In Figure 15, the red point indicates the

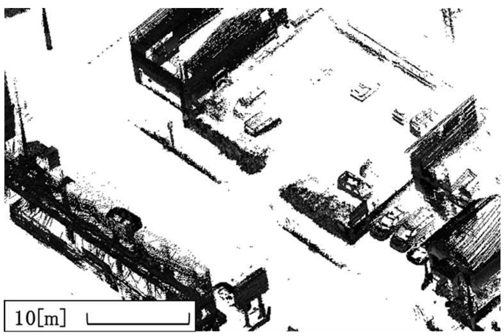
(a) Straight road

(b) Intersection

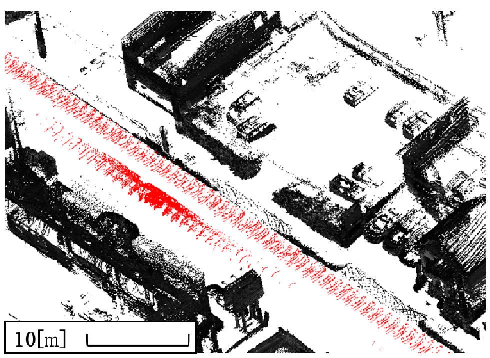
Figure 11. Photo of experimental environment.



(a) Photo

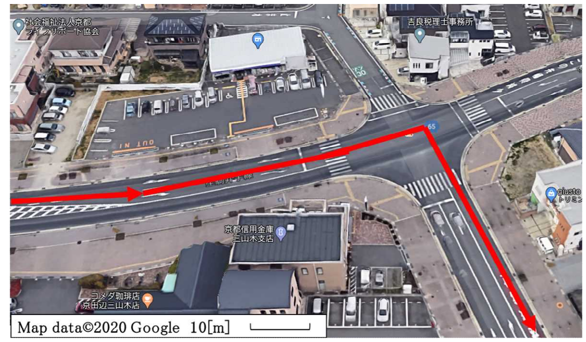


(b) Case 1

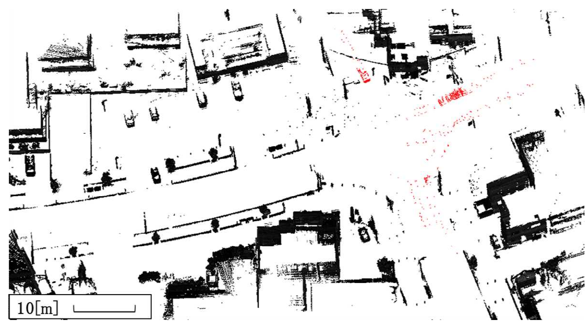


(c) Case 2

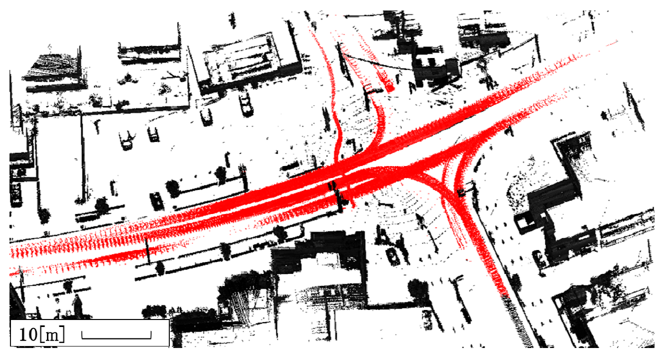
Figure 12. Mapping result of straight road area (bird's-eye view).



(a) Photo

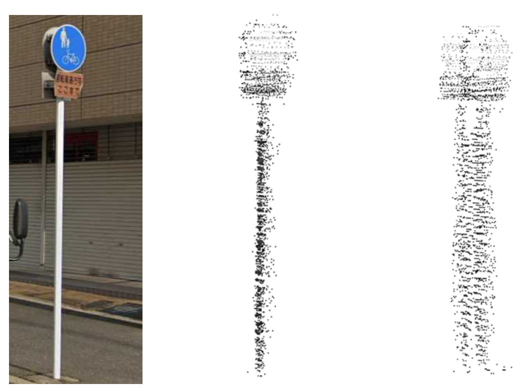


(b) Case 1



(c) Case 2

Figure 13. Mapping result of intersection area (bird's-eye view).



(a) Photo (b) Case 1 (c) Case 2

Figure 14. Mapping result of traffic sign.

start/goal position of the vehicle. The black, blue, and green lines indicate the movement paths of the vehicle in areas 1, 2, and 3, respectively. The broken-line circles indicate the locations, at which areas 1, 2, and 3 overlap.

Figure 16 shows photos of the start/goal position and intersections 1 and 2, which are shown in Figure 15. In the residential environment, there are three cars and three pedestrians. One of the three cars always follows the experimental vehicle.

For comparison, maps are generated in the following cases:

Case 1: NDT-SLAM-based single-session mapping (single map generation) through the distortion correction of

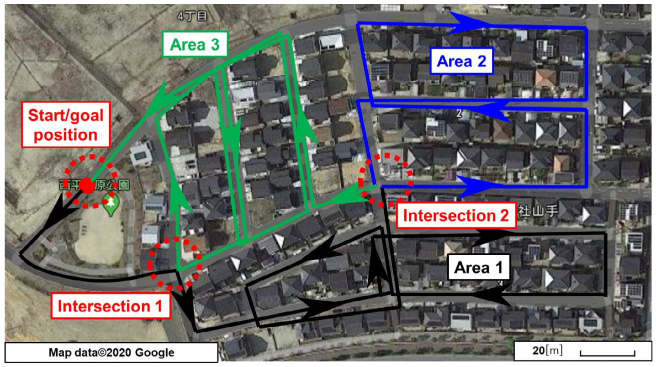


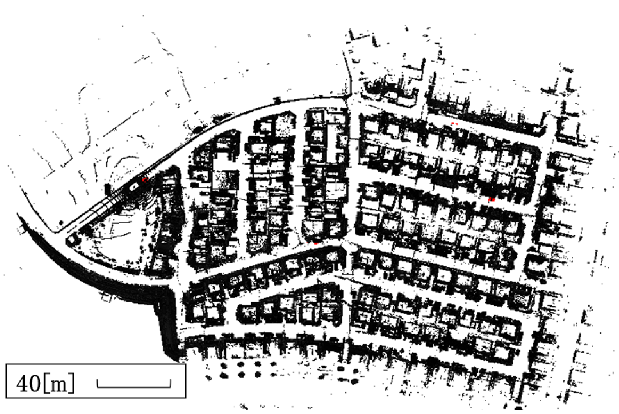
Figure 15. Movement path of vehicle (top view).

(a) Start/goal position

(b) Intersection 1

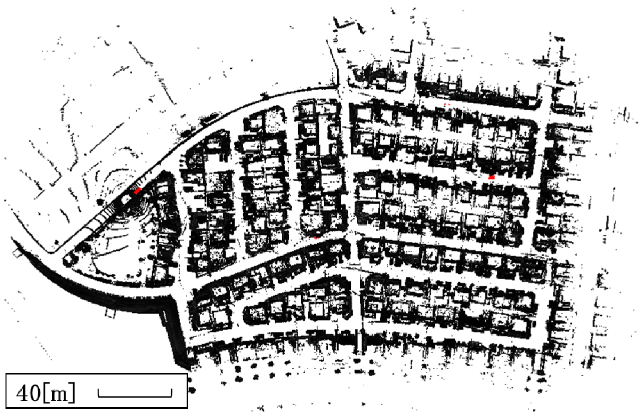
(c) Intersection 2

Figure 16. Photo of residential environment.



(a) Case 1

(b) Case 2



(c) Case 3

(d) Case 4

Figure 17. Mapping results (top view).

the LiDAR scan data and extraction of the static scan data from the LiDAR scan data;

Case 2: NDT-SLAM-based single-session mapping without using either method;

Case 3: NDT-Graph-SLAM-based single-session mapping using both methods;

Case 4: NDT-Graph-SLAM-based multisession mapping (submap generation and mapping) using both methods (proposed method).

For case 4, we split the recorded LiDAR scan data into three segments that are assumed to be created independently in the three areas (areas 1, 2, and 3) shown in Figure 15. We then generate and merge three submaps using the split LiDAR scan data. The experimental vehicle moves approximately 700 m, 600 m, and 700 m in areas 1, 2, and 3, respectively. These three areas overlap at the start/goal position and intersections 1 and 2 in Figure 15. Submaps 1 and 2 are firstly merged, and their enlarged submaps are further merged with submap 3.

Figure 17 shows the mapping results in cases 1–4, where the black and red dots indicate the static and moving scan data, respectively. In case 3, 2799 revisit nodes are detected, and the map generated by NDT SLAM is modified. In case 4, the numbers of detected encounter nodes are 284, 543, and 1486 for submaps 1, 2, and 3, respectively. 24 encounter nodes are detected when submap 1 is merged with submap 2. Then, 1287 encounter nodes are detected when the enlarged submaps are further merged with submap 3.

As seen in Figure 17, although the mapping performance in case 2 is the worst, the difference in the mapping performance in cases 1, 3, and 4 is unclear due to the small scale of the map. In SLAM, the worse the performance of the self-location of the vehicle, the worse the mapping performance. Therefore, to quantitatively evaluate the mapping performance, we obtain the estimate error in the vehicle self-location estimated by SLAM, where position information from the onboard GNSS/INS unit is used as the ground truth of the vehicle.

Table I shows the deviation between the start and goal positions of the vehicle. Table II also shows the Root-Mean-Square Error (RMSE) of the self-location in the entire movement path of the vehicle. It is concluded from these tables that case 3 (single-session NDT-Graph SLAM) and case 4 (multisession NDT-Graph SLAM) provide better results than cases 1 and 2 (single-session NDT SLAM) do.

In the experiment in the residential environment, moving objects, such as cars and pedestrians, are very few. An experiment in an urban road environment is further conducted to show the mapping performance of the proposed method in dynamic environments.

The movement path of the vehicle and photo of the environment are shown in Figures 18 and 19, respectively. The traveled distance of the experimental vehicle is about 2900 m, and the maximum speed of the vehicle is 40 km/h. In the road environment, there are 114 cars, 26 two-wheelers, and 37 pedestrians.

For comparison, maps are generated in the four above-mentioned cases. For case 4, we split the recorded sensor data into three segments that are assumed to be created independently in the three areas (areas 1, 2, and 3) shown in Figure 18. We then generate and merge three submaps using the split sensor data. The vehicle moves approximately 900 m, 1100 m, and 900 m in areas 1, 2, and 3, respectively. These three areas overlap at intersection 1 in Figure 18.

Submaps 1 and 2 are firstly merged, and their enlarged submaps are further merged with submap 3. In case 3, 306 revisit nodes are detected, and the map generated by NDT SLAM is modified. In case 4, the numbers of detected encounter nodes are zero, 39, and zero for submaps 1, 2, and 3, respectively, because areas 1 and 3 are straight roads. 24 encounter nodes are detected when submap 1 is merged with submap 2. Then, 977 encounter nodes are detected when the enlarged submaps are further merged with submap 3.

Figure 20 shows the mapping result, where the black and green dots indicate the static scan data extracted in areas 1 and 3, respectively. The red dot indicates the moving scan data. Tables III and IV show the self-location results of the vehicle, which are estimated by SLAM.

As seen in Figure 20, the tracks of cars remain in case 2 because we do not implement the algorithm that removes the moving scan data from the entire LiDAR scan data.

TABLE I. DEVIATION BETWEEN START AND GOAL POSITIONS OF VEHICLE IN RESIDENTIAL ENVIRONMENT.

TRUE	*CASE 1*	*CASE 2*	*CASE 3*	*CASE 4*
12.31 m	14.43 m	32.40 m	12.12 m	11.75 m

TABLE II. RMSE OF SELF-LOCATION OF VEHICLE IN RESIDENTIAL ENVIRONMENT.

CASE 1	*CASE 2*	*CASE 3*	*CASE 4*
1.48 m	9.86 m	1.00 m	0.99 m

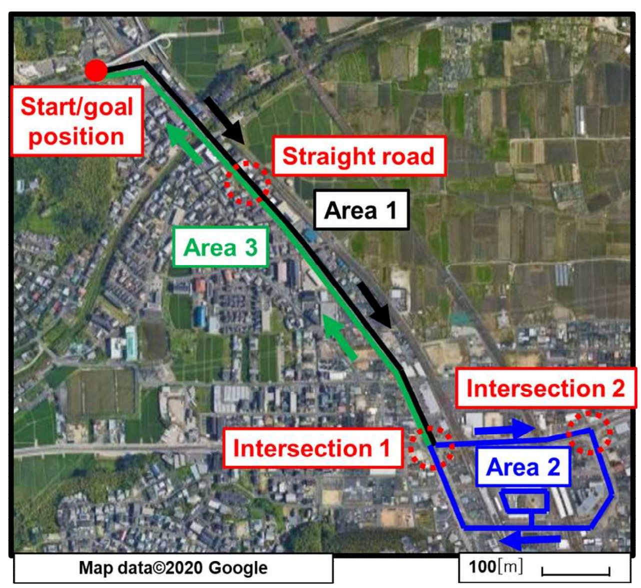
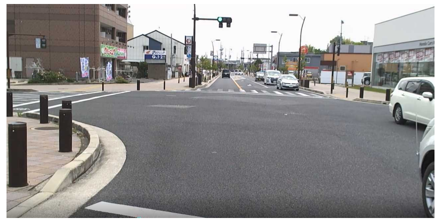


Figure 18. Moved path of vehicle (top view).

(a) Straight road

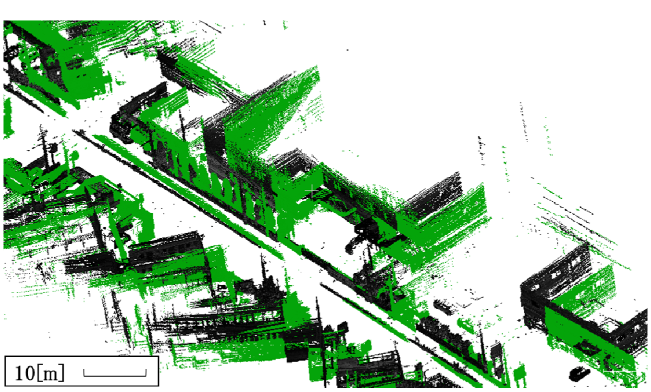


(b) Intersection 1

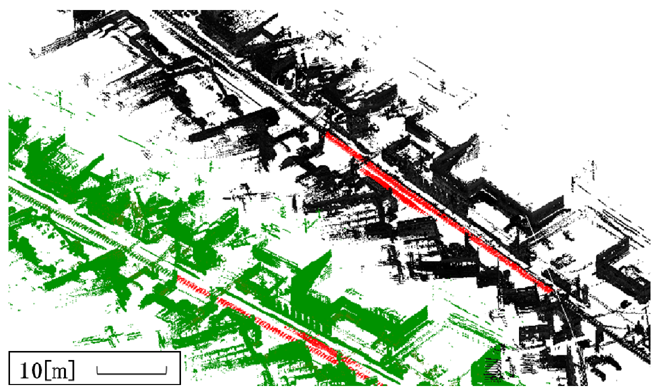
(c) Intersection 2

Figure 19. Photo of urban road environment.

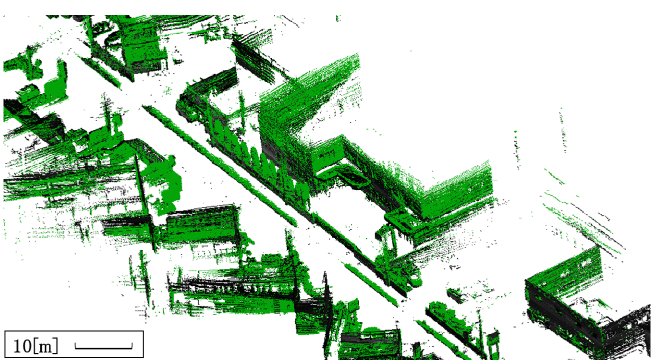
(a) Photo



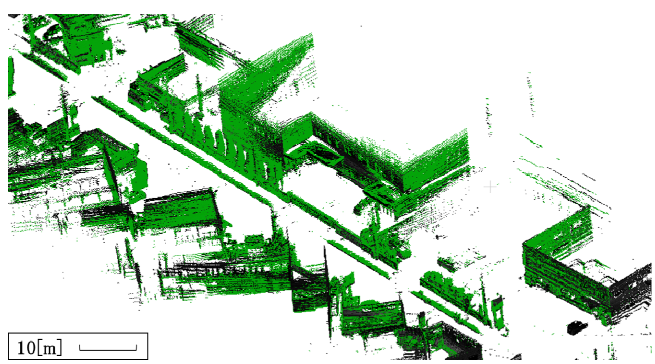
(b) Case 1



(c) Case 2



(d) Case 3



(e) Case 4

Figure 20. Mapping results (bird's-eye view).

TABLE III. DEVIATION BETWEEN START AND GOAL POSITIONS OF VEHICLE IN URBAN ROAD ENVIRONMENT

TRUE	*CASE 1*	*CASE 2*	*CASE 3*	*CASE 4*
3.50 m	17.34 m	126.19 m	6.10 m	3.38 m

TABLE IV. RMSE OF SELF-LOCATION OF VEHICLE IN URBAN ROAD ENVIRONMENT

CASE 1	*CASE 2*	*CASE 3*	*CASE 4*
5.95 m	35.95 m	9.86 m	3.23 m

Case 2 also causes a large drift in mapping due to the distortion of the LiDAR scan data and the accumulation error of NDT SLAM. The drift in case 1 is smaller than that in case 2 because the distortion of the LiDAR scan data is corrected in case 2. When the traveled distance of the vehicle is long, the accumulation error of NDT SLAM often deteriorates the performance of loop closure detection in Graph SLAM. For this reason, as seen in Table IV, the self-location error in case 3 (single-session NDT-Graph SLAM) is worse than that in case 1 (single-session NDT SLAM). Case 4 (proposed method) provides the best performance because shortly traveled distances in submaps reduce the accumulation error of NDT SLAM.

VIII. CONCLUSION AND FUTURE WORK

This paper presented a method of LiDAR-based mapping and merging in GNSS-denied and dynamic environments using only an onboard scanning LiDAR. 3D point cloud mapping and merging were performed by integrating three previously proposed algorithms: distortion correction of LiDAR scan data, extraction of static scan data (removal of moving scan data) from the entire LiDAR scan data, and single-session and multisession mapping using NDT-Graph SLAM. The mapping performance was shown through experiments conducted in outdoor residential and urban road environments.

We are currently evaluating the proposed method by mapping various environments, including large-scale residential environments. Some improvements to the presented method are required. Since the distortion correction of the LiDAR scan data requires a great deal of computational time, Graphical Processing Unit (GPU) or Field-Programmable Gate Array (FPGA) must be utilized in real-time operations. In our method of moving-object detection, when, for example, cars slow down at an intersection, stop at a red light, or pause to turn left (or right), they are sometimes determined as static objects. Then, the LiDAR scan data that relate to cars partially remain on the environmental map. To address this problem, study on map update and maintenance is needed.

ACKNOWLEDGMENTS

This study was partially supported by the KAKENHI Grant #18K04062, the Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] M. Yamaji, S. Tanaka, M. Hashimoto, and K. Takahashi, “Point Cloud Mapping Using Only Onboard Lidar in GNSS Denied and Dynamic Environments,” Proc. of the 15th Int. Conf. on Systems (ICONS 2020), pp. 43–49, 2020.

[2] C. Cadena et al., “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age,” IEEE Trans. on Robotics, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] L. Wang, Y. Zhang, and J. Wang, “Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR,” IFAC-Papers OnLine, vol. 50, issue 1, pp. 276-281, 2017.

[4] B. Huang, J. Zhao, and J. Liu. “A Survev of Simultaneous Localization and Mapping,” eprint arXiv:1909.05214, 2019.

[5] 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.

[6] H. G. Seif and X. Hu, “Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry,” Engineering, vol. 2, pp. 159–162, 2016.

[7] K. Morita, M. Hashimoto, and K. Takahashi, “Point-Cloud Mapping and Merging Using Mobile Laser Scanner,” Proc. of the third IEEE Int. Conf. on Robotic Computing (IRC 2019), pp. 417–418, 2019.

[8] D. Schwesinger, A. Shariati, C. Montella, and J. Spletzer, “A Smart Wheelchair Ecosystem for Autonomous Navigation in Urban Environments,” Autonomous Robot, vol. 41, pp. 519–538, 2017.

[9] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving,” IEEE Trans. on Intelligent Vehicles, vol. 2, pp. 194–220, 2017.

[10] K. Inui, M. Morikawa, M. Hashimoto, and K. Takahashi, “Distortion Correction of Laser Scan Data from In-vehicle Laser Scanner Based on Kalman Filter and NDT Scan Matching,” Proc. of the 14th Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO), pp. 329–334, 2017.

[11] S. Sato, M. Hashimoto, M. Takita, K. Takagi, and T. Ogawa, “Multilayer Lidar-Based Pedestrian Tracking in Urban Environments,” Proc. of IEEE Intelligent Vehicles Symp. (IV2010), pp. 849–854, 2010.

[12] S. Kanaki et al., “Cooperative Moving-Object Tracking with Multiple Mobile Sensor Nodes -Size and Posture Estimation of Moving Objects Using In-Vehicle Multilayer Laser Scanner-,” Proc. of 2016 IEEE Int. Conf. on Industrial Technology (ICIT 2016), pp. 59–64, 2016.

[13] S. Hong, H. Ko, and J. Kim, “VICP: Velocity Updating Iterative Closest Point Algorithm,” Proc. of 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010), pp. 1893–1898, 2010.

[14] F. Moosmann and C. Stiller, “Velodyne SLAM,” Proc. of IEEE Intelligent Vehicles Symp. (IV2011), pp. 393–398, 2011.

[15] J. Zhang and A. Singh, “LOAM: Lidar Odometry and Mapping in Real-time,” Proc. of Robotics: Science and Systems, 2014.

[16] 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 (IROS 2003), pp. 2743–2748, 2003.

[17] P. J. Besl and N. D. McKay, “A Method of Registration of 3-D Shapes,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992.

[18] W. Wen, L. T. Hsu, and G. Zhang, “Performance Analysis of NDT-Based Graph SLAM for Autonomous Vehicle in Diverse Typical Driving Scenarios of Hong Kong,” Sensors, 2018.

[19] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A Tutorial on Graph-Based SLAM,” IEEE Intelligent Transportation Systems Magazine, pp. 31–43, 2010.

[20] M. Labbe and F. Michaud, “Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM,” Proc. of the 2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2014), 2014.

[21] J. McDonald et al., “6-DOF Multi-Session Visual SLAM Using Anchor Nodes,” Proc. of Int. Conf. on European Conf. on Mobile Robots (ECMR), pp. 69–76, 2011

[22] J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “3D Normal Distributions Transform Occupancy Maps: An Efficient Representation for Mapping in Dynamic Environments,” Int. J. of Robotics Research, vol.32, no.14, pp. 1627–1644, 2013.

[23] X. Ding, Y. Wang, H. Yin, L. Tang, and R. Xiong, “Multi-Session Map Construction in Outdoor Dynamic Environment,” Proc. of the 2018 IEEE Int. Conf. on Real-time Computing and Robotics (IRC 2018), pp. 384–389, 2018.

[24] S. Thrun, “Learning Occupancy Grid Maps with Forward Sensor Models,” Autonomous Robots, vol.15, pp. 111–127, 2003.

[25] M. Munaro, F. Basso, and E. Menegatti, “Tracking People within Groups with RGB-D Data,” Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012), pp. 2101–2107, 2012.

[26] R. B. Rusu and S. Cousins, “3D is Here: Point Cloud Library (PCL),” Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011), 2011.

[27] 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.

[28] O. Bengtsson and A. J. Baerveldt, “Robot Localization Based on Scan-Matching—Estimating the Covariance Matrix for the IDC Algorithm,” Robotics and Autonomous Systems, vol. 44, pp. 29–40, 2003.

[29] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A General Framework for Graph Optimization,” Proc. of 2011 IEEE Int. Conf. on Robotics and Automation, pp. 3607–3613, 2011.