Moving-object detection and tracking by scanning LiDAR mounted on motorcycle based on dynamic background subtraction

Shotaro Muro¹ · Ibuki Yoshida¹ · Masafumi Hashimoto² · Kazuhiko Takahashi²

Received: 11 April 2021 / Accepted: 26 July 2021 / Published online: 20 August 2021
© The Author(s) 2021

Abstract

This paper presents a method for moving-object detection and tracking (DATMO) in global navigation satellite systems (GNSS)-denied environments using a light detection and ranging (LiDAR) mounted on a motorcycle. Distortion in the scanning LiDAR data is corrected by estimating the pose (3D positions and attitude angles) of the motorcycle in a period shorter than the LiDAR scan period using normal distributions transform-based simultaneous localization and mapping (NDT-based SLAM) and the information from an inertial measurement unit (IMU) via the extended Kalman filter (EKF). The scan data of interest are extracted by subtracting the local environment map generated by NDT-based SLAM from the LiDAR scan data. Moving objects are detected from the scan data of interest using an occupancy grid method and are tracked with a Bayesian filter. Experimental results obtained from public road and university campus environments demonstrate the effectiveness of the proposed method.

Keywords Motorcycle · LiDAR · Moving-object detection and tracking · Dynamic background subtraction

1 Introduction

In the field of mobile robots and intelligent transportation systems (ITS), studies on advanced driver assistance systems (ADAS) and autonomous driving of robots and vehicles are being actively conducted. Detection and tracking (estimation of position, velocity, and size) of moving objects, such as pedestrians, cars, and motorcycles, using onboard sensors, such as cameras, light detection and rangings (LiDARs), and radars, are required to support ADAS and autonomous driving [1–3].

Most moving-object detection and tracking (DATMO) methods are applied to ADAS and autonomous driving for cars and trucks (four-wheeled vehicles) traveling on flat road surfaces. As is the case with four-wheeled vehicles, advanced rider assistance systems (ARAS) are required for

motorcycles, but only a few studies have covered DATMO using sensors mounted on motorcycles. In [4–8], only forward and/or rear vehicles were detected with radar [4, 5], single-beam LiDAR [6], and stereo camera [7, 8]; the rear vehicle detection at a blind spot assisted in safe lane changes for motorcycles, and forward vehicle detection helped avoid rear-end collisions. In [9], opposing traffic with vehicles that travel straight or turn left at a traffic intersection was detected and tracked with a motorcycle-mounted 2D LiDAR with a narrow field of view (FOV) to avoid collisions between the motorcycle and left-turning vehicles.

Compared with camera-based DATMO, LiDAR-based DATMO is robust to lighting conditions and require less computational time. Furthermore, the tracking accuracy of LiDAR-based DATMO is better than that of radar-based DATMO due to the higher spatial resolution of LiDAR. Therefore, in this paper, we focus on LiDAR-based DATMO.

Previous studies on LiDAR-based DATMO for ARAS [6, 9] utilized 2D LiDAR with narrow FOVs and were designed under the assumption that the motorcycle equipped with LiDAR travels straight. This assumption would cause track lost when the motorcycle performs large attitude changes, such as lane-change maneuvers and turn motions at traffic intersections. To address this problem, this paper presents

---

This work was presented in part at the 26th International Symposium on Artificial Life and Robotics (Online, January 21–23, 2021).

✉ Masafumi Hashimoto
mhashimo@mail.doshisha.ac.jp

¹ Graduate School of Doshisha University, Kyoto, Japan

² Doshisha University, Kyoto, Japan

LiDAR-based DATMO using a scanning 3D LiDAR with 360° FOV that is mounted on a motorcycle.

LiDAR-based surrounding environment recognition methods, including DATMO and simultaneous localization and mapping (SLAM), require accurate mapping of LiDAR scan data captured in a sensor coordinate frame onto a world coordinate frame using the self-pose (position and attitude angle) information of the motorcycle. The LiDAR obtains range measurements by scanning LiDAR beams. Thus, when the motorcycle exhibits large changes in pose (position and attitude), the entire scan data cannot be obtained at the same pose of the motorcycle. Therefore, if the entire scan data obtained within one scan are mapped onto the world coordinate frame using information about the pose of the motorcycle at a single point in time, distortion arises in mapping [10, 11], and tracking errors occur.

For accurate mapping under large pose changes in global navigation satellite systems (GNSS)-denied environments, such as urban street canyons in which the accuracy of the motorcycle's self-pose using GNSS significantly deteriorates, our previous work [12] proposed a distortion correction method for LiDAR scan data using normal distributions transform (NDT)-based SLAM and information from an inertial measurement unit (IMU).

Moving-object detection is usually performed by extracting scan data originating from moving objects, that is, removing scan data originating from static objects, from the entire LiDAR scan data using the occupancy grid method [13]. However, in practical environments, LiDAR noises and outliers frequently cause false tracking through erroneous detection of static objects as moving objects. An effective approach to reducing false tracking is DATMO based on environment map subtraction [14]. In this method, a 3D point cloud environment map built by LiDAR-based SLAM [15] is prepared in advance. The scan data of interest are extracted by subtracting the environment map from the current LiDAR scan data, and scan data related to moving objects from the scan data of interest are detected and tracked.

Although environment map subtraction can improve tracking performance, it requires an environment map in advance. To address this problem, in this paper, a local environment map is sequentially built using NDT-based SLAM, and the scan data of interest are extracted by subtracting the local environment map from the current LiDAR scan data. This extraction method is called dynamic background subtraction (DBS)-based extraction, which will enable accurate DATMO in first-visit environments.

In this paper, the performance of DATMO in conjunction with DBS-based extraction is demonstrated through experimental results from public road and university campus road environments. The rest of this paper is organized as follows. Section 2 describes the experimental system.

Section 3 briefly explains NDT-based SLAM. Section 4 explains the distortion correction in LiDAR scan data, and Sect. 5 presents the DATMO using DBS-based extraction. Finally, Sect. 6 explains the experiments conducted to show the performance of our method, followed by the conclusions in Sect. 7.

2 Experimental system

Figure 1 shows an overview of our experimental motorcycle (Honda, Gyro Canopy). The top part of the motorcycle is equipped with a 32-layer LiDAR (Velodyne, HDL-32E) and an IMU (Xsens, MTi-300). 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°. LiDAR acquires 384 measurements (the 3D position of the object and reflection intensity) every 0.55 ms (at 2° horizontal angle increments). The period needed by the LiDAR beam to complete one rotation (360°) in the horizontal direction is 100 ms, and approximately 70,000 measurements are thus acquired in one rotation.

The IMU outputs the attitude angle (roll and pitch angles) and angular velocity (roll, pitch, and yaw angular velocities) every 10 ms. The errors in attitude angle and angular velocity are less than $\pm 0.3^\circ$ and $\pm 0.2^\circ/\text{s}$, respectively.

Fig. 1 Experimental system

In this paper, one rotation of the LiDAR beam in the horizontal direction (360°) is called one scan, while the data obtained from this scan is called scan data. The scan period (100 ms) of the LiDAR is denoted as τ , and the scan-data observation period (0.55 ms) as $\Delta\tau$. The observation period (10 ms) of the IMU is denoted as $\Delta\tau_{IMU}$, which means the IMU data are obtained 10 times in one scan of the LiDAR ($\tau = 10\Delta\tau_{IMU}$), while the LiDAR scan data are

$$\begin{pmatrix} p_i \\ 1 \end{pmatrix} = T(X) \begin{pmatrix} p_{bi} \\ 1 \end{pmatrix} \quad (1)$$

where $X = (x, y, z, \phi, \theta, \psi)^T$ is the pose of the motorcycle. $(x, y, z)^T$ and $(\phi, \theta, \psi)^T$ are the 3D position and attitude angle (roll, pitch, and yaw angles), respectively, of the motorcycle in Σ_W . $T(X)$ is the homogeneous transformation matrix, and it is represented as follows:

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

obtained 18 times within the observation period of the IMU ($\Delta\tau_{IMU} = 18\Delta\tau$).

3 NDT-based SLAM

Figure 2 shows the flow of our DATMO. The LiDAR scan data are mapped from the motorcycle's coordinate frame Σ_b onto the world coordinate frame Σ_W using the self-pose (3D position and attitude angle) information of the motorcycle. The self-pose of the motorcycle needs to be accurate. NDT-based SLAM [16] is used to estimate the self-pose in GNSS-denied environments.

For the i -th ($i = 1, 2, \dots, n$) measurement in the scan data, the position vector in Σ_b is defined as $p_{bi} = (x_{bi}, y_{bi}, z_{bi})^T$, and that in Σ_W as $p_i = (x_i, y_i, z_i)^T$. The following relation is then given:

The scan data obtained at the current time are called the current scan data, and the scan data obtained prior to the current time are called the local environment map. The pose X is calculated by matching the current scan data with the local environment map. The current scan data are mapped onto Σ_W using X via Eq. (1), and the local environment map is then updated.

The local environment map should consist only of scan data related to static objects (static scan data), such as building walls, utility poles, and trees. Therefore, as shown in Fig. 2, the static scan data, which are removed by the DBS-based extraction method (Subsection 5.1) and extracted by the occupancy grid method (Subsection 5.2), are merged with the local environment map.

Fig. 2 Flow of moving-object tracking

Fig. 3 Notation related to motorcycle motion

4 Distortion correction of LiDAR scan data

4.1 Motion and measurement models

As shown in Fig. 3, the linear velocity of the motorcycle in Σ_b is denoted as V_b (velocity in the x_b -axis direction), and the angular velocities around the x_b , y_b , and z_b axes are denoted as $\dot{\phi}_b$, $\dot{\theta}_b$, and $\dot{\psi}_b$, respectively.

If the motorcycle is assumed to move at nearly constant linear and angular velocities, a motion model can be derived by

$$\begin{pmatrix} x(t+1) \\ y(t+1) \\ z(t+1) \\ \phi(t+1) \\ \theta(t+1) \\ \psi(t+1) \\ V_b(t+1) \\ \dot{\phi}_b(t+1) \\ \dot{\theta}_b(t+1) \\ \dot{\psi}_b(t+1) \end{pmatrix} = \begin{pmatrix} x(t) + a_1(t)\cos\theta(t)\cos\psi(t) \\ y(t) + a_1(t)\cos\theta(t)\sin\psi(t) \\ z(t) - a_1(t)\sin\theta(t) \\ \phi(t) + a_2(t) + \{a_3(t)\sin\phi(t) + a_4(t)\cos\phi(t)\}\tan\theta(t) \\ \theta(t) + a_3(t)\cos\phi(t) - a_4(t)\sin\phi(t) \\ \psi(t) + \{a_3(t)\sin\phi(t) + a_4(t)\cos\phi(t)\}\frac{1}{\cos\theta(t)} \\ V_b(t) + \tau w_{V_b} \\ \dot{\phi}_b(t) + \tau w_{\dot{\phi}_b} \\ \dot{\theta}_b(t) + \tau w_{\dot{\theta}_b} \\ \dot{\psi}_b(t) + \tau w_{\dot{\psi}_b} \end{pmatrix} \quad (2)$$

where (x, y, z) and (ϕ, θ, ψ) are the 3D position and attitude angle (roll, pitch, and yaw angles), respectively of the motorcycle. $(\dot{\phi}_b, \dot{\theta}_b, \dot{\psi}_b)$ are the angular velocities (roll, pitch, and yaw velocities) of the motorcycle, and $(w_{V_b}, w_{\dot{\phi}_b}, w_{\dot{\theta}_b}, w_{\dot{\psi}_b})$ are the acceleration disturbances. $a_1 = V_b\tau + \tau^2 w_{V_b}/2$, $a_2 = \dot{\phi}_b\tau + \tau^2 w_{\dot{\phi}_b}/2$, $a_3 = \dot{\theta}_b\tau + \tau^2 w_{\dot{\theta}_b}/2$, and $a_4 = \dot{\psi}_b\tau + \tau^2 w_{\dot{\psi}_b}/2$.

Equation (2) is expressed by the following vector form:

$$\xi(t+1) = f[\xi(t), w, \tau] \quad (3)$$

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

The attitude angle (roll and pitch angles) and angular velocity (roll, pitch, and yaw angular velocities) of the motorcycle obtained by the IMU at time $t\tau_{IMU}$ are denoted as $z_{IMU}(t)$. The measurement model is then

$$\left. \begin{aligned} \hat{\xi}^{(k/k-1)}(t-1) &= f[\hat{\xi}^{(k-1)}(t-1), 0, \Delta\tau_{IMU}] \\ \mathbf{I}^{(k/k-1)}(t-1) &= \mathbf{F}(t-1)\mathbf{I}^{(k-1)}(t-1)\mathbf{F}^T(t-1) + \mathbf{G}(t-1)\mathbf{Q}\mathbf{G}^T(t-1) \end{aligned} \right\} \quad (6)$$

$$z_{IMU}(t) = \mathbf{H}_{IMU}\xi(t) + \Delta z_{IMU}(t) \quad (4)$$

where Δz_{IMU} is the sensor noise, and $\mathbf{H}_{IMU}$ is the following measurement matrix:

$$\mathbf{H}_{IMU} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The pose of the motorcycle obtained at $t\tau$ using NDT scan matching is denoted as $z_{NDT}(t) \equiv \hat{X}(t)$. The measurement model is then

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

where Δz_{NDT} is the measurement noise, and $\mathbf{H}_{NDT}$ is the following measurement matrix:

$$\mathbf{H}_{NDT} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4.2 Distortion correction

Figure 4 shows the flow of the distortion correction in LiDAR scan data [12]. The scan period τ of the LiDAR is 100 ms, the observation period $\Delta\tau_{IMU}$ of the IMU is 10 ms, and the scan data observation period $\Delta\tau$ is 0.55 ms. When the scan data are mapped onto Σ_w using the pose of the motorcycle, which is calculated for every LiDAR scan period, distortion appears in the mapping of LiDAR scan data onto Σ_w . Therefore, the distortion in the LiDAR scan data is corrected by estimating the pose of the motorcycle using extended Kalman filter (EKF) for every scan-data observation period $\Delta\tau$.

The IMU data are obtained 10 times per LiDAR scan ($\tau = 10\Delta\tau_{IMU}$). The state estimate of the motorcycle and its error covariance obtained using the EKF at time $(t-1)\tau + (k-1)\Delta\tau_{IMU}$, where $k=1-10$, are denoted as $\hat{\xi}^{(k-1)}(t-1)$ and $\mathbf{I}^{(k-1)}(t-1)$, respectively.

From these quantities, the one-step prediction algorithm by the EKF gives the state prediction $\hat{\xi}^{(k/k-1)}(t-1)$ and the error covariance $\mathbf{I}^{(k/k-1)}(t-1)$ at $(t-1)\tau + k\Delta\tau_{IMU}$ by

Fig. 4 Flow of distortion correction

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

At $(t-1)\tau + k\Delta\tau_{IMU}$, the attitude angle and angular velocity z_{IMU} of the motorcycle are observed with the IMU. Then, the EKF estimation algorithm gives the state estimate $\hat{\xi}^{(k)}(t-1)$ and its error covariance $\Gamma^{(k)}(t-1)$ as follows:

$$\left. \begin{aligned} \hat{\xi}^{(k)}(t-1) &= \hat{\xi}^{(k/k-1)}(t-1) + K\{z_{IMU} - H_{IMU}\hat{\xi}^{(k/k-1)}(t-1)\} \\ \Gamma^{(k)}(t-1) &= \Gamma^{(k/k-1)}(t-1) - KH_{IMU}\Gamma^{(k/k-1)}(t-1) \end{aligned} \right\} \quad (7)$$

where

$$\left. \begin{aligned} K &= \Gamma^{(k/k-1)}(t-1)H_{IMU}^T S^{-1} \\ S &= H_{IMU}\Gamma^{(k/k-1)}(t-1)H_{IMU}^T + R_{IMU} \end{aligned} \right\}$$

and, R_{IMU} is the covariance matrix of the sensor noise Δz_{IMU} .

In the state estimate $\hat{\xi}^{(k)}(t-1)$, the elements related to the pose of the motorcycle ($x, y, z, \phi, \theta, \psi$) are denoted as $\hat{X}^{(k)}(t-1)$. Since the observation period $\Delta\tau_{IMU}$ of the IMU is 10 ms, and the scan data observation period $\Delta\tau$ is 0.55 ms, the LiDAR scan data are obtained 18 times within the IMU observation period ($\Delta\tau_{IMU} = 18\Delta\tau$).

With use of the pose estimates $\hat{X}^{(k-1)}(t-1)$ and $\hat{X}^{(k)}(t-1)$ obtained at $(t-1)\tau + (k-1)\Delta\tau_{IMU}$ and $(t-1)\tau + k\Delta\tau_{IMU}$, respectively, the pose of the motorcycle $\hat{X}^{(k-1)}(t-1, j)$ at $(t-1)\tau + (k-1)\Delta\tau_{IMU} + j\Delta\tau$, where $j = 1-18$, is calculated via the linear interpolation:

$$\hat{X}^{(k-1)}(t-1, j) = \hat{X}^{(k-1)}(t-1) + \frac{\hat{X}^{(k)}(t-1) - \hat{X}^{(k-1)}(t-1)}{\Delta\tau_{IMU}} j\Delta\tau \quad (8)$$

With use of Eq. (1) and the pose prediction $\hat{X}^{(k-1)}(t-1, j)$, the scan data $p_{bi}^{(k-1)}(t-1, j)$ in Σ_b obtained at $(t-1)\tau + (k-1)\Delta\tau_{IMU} + j\Delta\tau$ can be transformed to $p_i^{(k-1)}(t-1, j)$ in Σ_w as follows:

$$\begin{pmatrix} p_i^{(k-1)}(t-1, j) \\ 1 \end{pmatrix} = T(\hat{X}^{(k-1)}(t-1, j)) \begin{pmatrix} p_{bi}^{(k-1)}(t-1, j) \\ 1 \end{pmatrix} \quad (9)$$

Since the IMU data are obtained 10 times per LiDAR scan ($\tau = 10\Delta\tau_{IMU}$), the time $t\tau$ is equal to $(t-1)\tau + 10\Delta\tau_{IMU}$. With use of the pose estimate $\hat{X}^{(10)}(t-1)$ of the motorcycle at $t\tau$, the scan data $p_i^{(k-1)}(t-1, j)$ in Σ_w at $(t-1)\tau + (k-1)\Delta\tau_{IMU} + j\Delta\tau$ are transformed to the scan data $P_{bi}^*(t)$ in Σ_b at $t\tau$ by

$$\begin{pmatrix} P_{bi}^*(t) \\ 1 \end{pmatrix} = T(\hat{X}^{(10)}(t-1))^{-1} \begin{pmatrix} p_i^{(k-1)}(t-1, j) \\ 1 \end{pmatrix} \quad (10)$$

In such way, the corrected scan data $P_b^*(t) = \{p_{b1}^*(t), p_{b2}^*(t), \dots, p_{bn}^*(t)\}$ within one scan (LiDAR

beam rotation of 360° in a horizontal plane) are obtained and used as the new input scan data for the scan matching to calculate the pose z_{NDT} of the motorcycle at $t\tau$. Then, the EKF estimation algorithm is used to calculate the state estimate $\hat{\xi}(t)$ and its error covariance $\Gamma(t)$ of the motorcycle at $t\tau$ as follows:

$$\left. \begin{aligned} \hat{\xi}(t) &= \hat{\xi}^{(10)}(t-1) + K(t)\{z_{NDT}(t) - H_{NDT}\hat{\xi}^{(10)}(t-1)\} \\ \Gamma(t) &= \Gamma^{(10)}(t-1) - K(t)H_{NDT}\Gamma^{(10)}(t-1) \end{aligned} \right\} \quad (11)$$

where

$$\left. \begin{aligned} K(t) &= \Gamma^{(10)}(t-1)H_{NDT}^T S^{-1}(t) \\ S(t) &= H_{NDT}\Gamma^{(10)}(t-1)H_{NDT}^T + R_{NDT} \end{aligned} \right\}$$

and, R_{NDT} is the covariance matrix of Δz_{NDT} .

The corrected scan data $P_b^*(t)$ are mapped onto Σ_w using the pose estimate calculated by Eq. (11), and the distortion in the LiDAR scan data can then be removed.

5 Moving-object detection and tracking

5.1 Subtraction of scan data

Figure 5 shows an overview of the DBS-based extraction method. For moving-object tracking, scan data related to static objects (static scan data) have to be removed and those related to moving objects (moving scan data) have to be extracted from the entire LiDAR scan data. To remove as much static scan data as possible from the entire LiDAR scan data, we subtract the local environment map from the current scan data.

As the local environment map and current scan data contain a large amount of scan data, they are both downsized using a voxel grid filter. Here, the block for the voxel grid filter is a cube with a side length of 0.2 m.

5.2 Moving-object detection

The scan data extracted using the DBS-based method are mapped onto a grid map. Here, the cell is a square with a side length of 0.3 m. A cell that contains scan data is called an occupied cell. For the moving scan data, the time needed to occupy the same cell is short (less than 0.7 s in this paper), whereas for the static scan data, the time is long (at least 0.7 s). Therefore, with use of the occupancy grid method, which is based on the cell occupancy time [17], 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

Fig. 5 Dynamic background subtraction method (top view)

because of obstructions are defined as unknown cells, and their cell occupancy time is not counted.

Since scan data related to an object usually occupy multiple cells, adjacent occupied cells are clustered. Then, clustered moving cells (static cells) are obtained as moving-cell group (or static-cell group).

As the motorcycle moves, the LiDAR FOV also moves in Σ_w . In the occupancy grid method, which is based on the cell occupancy time, even if an object that newly enters the LiDAR FOV is a static object, it is misdetected as a moving cell because the cell occupancy time is short. To address this problem, new-observation cells are defined on the grid map, which correspond to the new FOV of the LiDAR. The time of cells entering the LiDAR FOV (T_{NC}) and the cell occupancy time (T_{OC}) are counted, and the occupancy time rate (α) is calculated by $\alpha = T_{OC}/T_{NC}$. Cells in which α is 10% or more are determined to be new-observation cells and then considered moving cells. This can reduce the false detection of static objects newly entering the LiDAR FOV as moving objects.

In our previous work [14], an advanced environment map with a dense point cloud could be used for the environment map subtraction method, and it resulted in accurate extraction of the moving scan data. On the other hand, this paper generates the local environment map by NDT-based SLAM. Here, the scan data in the local environment map are sparser than those in the advanced environment map, especially in the front of the motorcycle and the occlusion areas. For this

reason, when subtracting the local environment map from the current scan data, static scan data are also extracted in a sparse state. If the scan data, which are sparsely extracted by the DBS-based method, are mapped onto a grid map, they may be erroneously determined as a moving cell.

To address this problem, the scan data removed by the DBS-based method are also mapped onto the grid map as static cells. As a result, sparse static scan data that tend to be moving cells and static scan data that are removed by the DBS-based method are both mapped onto the grid map. Neighboring cells containing these static scan data are clustered, and the cell group is then determined to be a static-cell group. As a result, the sparse static scan data are correctly determined as static scan data by the occupancy grid method.

Fig. 6 Cuboid around tracked object (car)

Fig. 7 Observed vehicle size. The red squares and the green arrow indicate the moving cells and the vehicle heading direction, respectively. The blue rectangle and circle indicate the observed size and centroid, respectively

5.3 Moving-object tracking

The shape of the moving object is represented by a cuboid with a width W , length L , and height H , as shown in Fig. 6.

An $X_v Y_v$ -coordinate frame is defined in Fig. 7, on which the Y_v axis aligns with the heading of a tracked object. From the clustered moving cells (moving-cell group), the width W_{meas} and length L_{meas} are measured. When a moving object is perfectly visible, its size can accurately be estimated from the measurements W_{meas} and L_{meas} . In contrast, when it is partially occluded by other objects, its size is incorrectly estimated. Therefore, the size of a partially visible object is estimated using the following equation:

$$\begin{cases} W(t) = W(t-1) + G(W_{\text{meas}}(t) - W(t-1)) \\ L(t) = L(t-1) + G(L_{\text{meas}}(t) - L(t-1)) \end{cases} \quad (12)$$

where G is the filter gain, given by $G = 1 - \sqrt[10]{1 - \beta}$ [17]. The reliabilities of the current measurements of W_{meas} and L_{meas} increase with the value of β . A surrounding vehicle is assumed to pass at 60 km/h in front of the motorcycle. After the vehicle enters the LiDAR FOV, we aim to estimate the size correctly with the probability of 99% ($\beta = 0.99$) within 10 scans (1 s) of the LiDAR. The filter gain can then be determined as follows:

$$G = \begin{cases} 1 - \sqrt[10]{1 - 0.99} & \text{for } t \leq 10 \\ 1 - \sqrt[10]{1 - 0.99} = 0.369 & \text{for } t > 10 \end{cases} \quad (13)$$

The height of the moving-cell group is used as the height estimate H .

The centroid position (x, y) of the rectangle estimated from Eq. (12) is used as a position measurement of the moving object to estimate the position and velocity of the moving object in Σ_w using the Kalman filter [18]. In the use of the Kalman filter, the object is assumed to be moving at an approximately constant velocity.

In object tracking in crowded environments, data association (that is, one-to-one or one-to-many matching of tracked objects and moving-cell groups) is required. A rule-based data association method [18] is used to accurately match

(a) Overall

(b) Intersection

Fig. 8 Photo of environment 1 (top view). The red line indicates the movement path of the ego-vehicle (motorcycle)

Fig. 9 DATMO result at 158 s in environment 1 (top view)

multiple moving objects with multiple moving-cell groups in crowded environments.

The number of moving objects in the LiDAR FOV changes over time. Moving objects enter and exit the LiDAR FOV, and they interact with and become occluded by other objects in the environments. To handle such conditions, a rule-based data handling method including track initiation and termination [18] is implemented.

6 Experimental results

An experiment is conducted on a public road environment (environment 1), as shown in Fig. 8a. The maximum speed of the motorcycle is 40 km/h, and the distance traveled by the motorcycle is 1200 m. On the road, there are 18 pedestrians, 15 two-wheeled vehicles, and 38 cars.

Figure 9 shows the DATMO results in the intersection shown in Fig. 8b. The black dotted line indicates the movement path of the motorcycle. The light blue rectangle indicates the estimated size of the moving object, and the light blue line indicates the moving direction of the moving object obtained from the velocity estimate. The blue (red) dots indicate the scan data removed (extracted) from the LiDAR scan data using the DBS-based method. Figure 10 shows

(a) Roll (black) and pitch (red) angles

(b) Roll (black), pitch (red), and yaw (light blue) angular velocities

Fig. 10 Attitude angle and angular velocity of the ego-vehicle in environment 1

the attitude angles and angular velocities of the motorcycle moving in the intersection. The size of the moving objects shown in Fig. 9 is estimated based on LiDAR scan data at 150–158 s.

When the motorcycle turns left at the intersection, the maximum roll angle is 10° , and the maximum roll angular velocity is $14.5^\circ/\text{s}$. Figure 9 indicates that even when the motorcycle attitude changes significantly by turning left, the static data originating from the building walls and the stopped car are removed, and the moving objects are tracked.

The motorcycle is moved three times on the road shown in Fig. 8a. The total number of moving objects is 211 (133 cars, 28 two-wheelers, and 50 pedestrians). The tracking performance is compared in the following cases.

Case 1: tracking with distortion correction and DSM-based extraction (proposed method)

Case 2: tracking with distortion correction and without DSM-based extraction

Case 3: tracking with DSM-based extraction and without distortion correction

Case 4: Tracking with neither method

Table 1 shows the tracking result; untracking means failed tracking of moving object, and false tracking means tracking of static objects.

Since the experiment is conducted on a public road, the motorcycle attitude is significantly changed only when turning at intersections. Therefore, to investigate tracking performance when the motorcycle experiences a large attitude change, another experiment is conducted in our university

Table 1 Total number of correct and incorrect tracking in environment 1

	Correct tracking	Untracking	False tracking
Case 1	208	3	19
Case 2	208	3	36
Case 3	206	5	33
Case 4	207	4	50

Fig. 11 Photo of environment 2 (bird's-eye view). The red line indicates the movement path of the ego-vehicle (motorcycle)

(a) Roll (black) and pitch (red) angles

(b) Roll (black), pitch (red), and yaw (light blue) angular velocities

Fig. 12 Attitude angle and angular velocity of the ego-vehicle in environment 2 **Fig. 13** DATMO result at 86 s in environment 2 (top view)**Table 2** Total number of correct and incorrect tracking in environment 2

	Correct tracking	Untracking	False tracking
Case 1	236	1	3
Case 2	237	0	6
Case 3	227	10	14
Case 4	232	5	27

Fig. 14 Photo of guard pipe in environment 1

campus (environment 2). In this experiment, the motorcycle frequently moves in a zigzag path. Figure 11 shows the movement path of the motorcycle. The distance traveled by the motorcycle is 500 m, and the maximum speed is 30 km/h. On the road, there are 35 pedestrians and two cars. The attitude angle and angular velocity of the motorcycle are shown in Fig. 12.

Figure 13 shows the tracking results of the moving objects in environment 2. The size of the moving objects shown in this figure is estimated based on LiDAR scan data at 83–86 s. Therefore, the moving objects can be tracked even when the motorcycle attitude changes significantly. The motorcycle is moved five times on the road. The total number of moving objects is 237 (10 cars and 227 pedestrians). Table 2 shows the tracking result in environment 2.

The comparison between cases 1 and 2 (or cases 3 and 4) in Tables 1 and 2 shows that the DBS-based extraction method reduces the instances of false tracking. In addition, the comparison between cases 1 and 3 (or cases 2 and 4) indicates that the distortion correction of the LiDAR scan data reduces untracking. The proposed method (case 1) therefore has better tracking performance than the three other cases.

As seen in Tables 1 and 2, false tracking occurs more often in environment 1 than in environment 2. In environment 1, many guard pipes (guard fences), which are composed of thin beams (Fig. 14), stand on both sides of the road. The motorcycle drives on the left lane of the two-lane road, and the motorcycle-mounted LiDAR is far from the guard pipes located on the right side of the road. Since the vertical spatial resolution of the LiDAR is sparse, the right-side guard pipes are intermittently observed by the LiDAR and misrecognized as moving objects. Consequently, false tracking occurs more often in environment 1 than in environment 2.

In the experiments, LiDAR scan data are recorded, and DATMO is executed offline by a computer. The specifications of the computer are as follows: Windows 10 Pro OS,

Table 3 Processing time of DATMO in environment 1 (ms)

	NDT SLAM and distortion correction	DSM-based extraction	Others	Total
Case 1	673	364	58	1095
Case 2	671		72	743
Case 3	681	380	74	1135
Case 4	681		69	750

Table 4 Processing time of DATMO in environment 2 (ms)

	NDT SLAM and distortion correction	DSM-based extraction	Others	Total
Case 1	1116	946	61	2123
Case 2	1115		76	1191
Case 3	1242	1044	95	2381
Case 4	1249		82	1331

Fig. 15 Road view in environment 2

Intel(R) Core (TM) i7-7700 K @4.20 GHz CPU, 16 GB RAM, and C++ software language. The point cloud library (PCL) [19] is used for NDT-based SLAM. Tables 3 and 4 show the processing time (mean time) of DATMO in environments 1 and 2, respectively.

Although the distortion correction method is not utilized in cases 3 and 4, the processing time for NDT-based SLAM and distortion correction in cases 3 and 4 is almost the same as that in cases 1 and 2. This means that NDT-based SLAM requires long computational time. In addition, as shown in Tables 3 and 4, the processing time in environment 2 is longer than that in environment 1. The road in environment 2 is narrower than that in environment 1, and many trees are planted on both sides of the road in environment 2, as

shown in Fig. 15. The amount of LiDAR scan data captured in environment 2 is therefore larger than that in environment 1. Consequently, the processing time in environment 2 is longer than that in environment 1.

7 Conclusion

This paper presented a DATMO method that used a motorcycle-mounted scanning LiDAR. The distortion in scanning LiDAR data was corrected, and the self-pose information and local environment map were obtained using NDT-based SLAM in GNSS-denied environments. Moving objects were detected and then tracked by comparing the current LiDAR scan data with the local environment map. The performance of the proposed DATMO method was demonstrated through experiments conducted in public road and university campus road environments.

We are currently evaluating the proposed method through experiments in various environments, including urban city, mountainous, and uneven terrain environments. In addition, since the proposed method requires long computational time, as shown in Tables 3 and 4, we aim to reduce the computational time by optimizing the program code and using a graphical processing unit (GPU) for real-time operations.

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Mukhtar A, Xia L, Tang TB (2015) Vehicle detection techniques for collision avoidance systems: a review. *IEEE Trans Intell Transp Syst* 16:2318–2338
2. Marti E, Perez J, Miguel MA, Garcia F (2019) A review of sensor technologies for perception in automated driving. *IEEE Intell Transp Syst Mag* 11:94–108
3. Muller FP (2017) Survey on ranging sensors and cooperative techniques for relative positioning of vehicles. *Sensors* 17:271

4. Motorcycle Rider Safety (2021) <https://www.bosch.com/stories/motorcycle-rider-safety>. Accessed 16 Mar 2021
5. Amodio A, Panzani G, Savaresi SM (2017) Design of a lane change driver assistance system with implementation and testing on motorbike. In: Proceedings of IEEE Intelligent Vehicles Symposium, pp 947–952
6. Jeon W, Rajamani R (2018) Rear vehicle tracking on a bicycle using active sensor orientation control. *IEEE Trans Intell Transp Syst* 19:2638–2649
7. Gil G, Savino G, Piantini S, Pierini M (2018) Motorcycles that see: multifocal stereo vision sensor for advanced safety systems in tilting vehicles. *Sensors* 18:295
8. Kumarasamy G, Prakash NK, Mohan PS (2015), Rider assistance system with an active safety mechanism. In: Proceedings of 2015 IEEE International Conference on Computational Intelligence, pp 163–168
9. Xie Z, Rajamani R (2019) On-bicycle vehicle tracking at traffic intersections using inexpensive low-density lidar. In: Proceedings of American Control Conference, pp 953–958
10. Hong S, Ko H, Kim J (2010) VICP: velocity updating iterative closest point algorithm. In: Proceedings of 2010 IEEE International Conference on Robotics and Automation, pp 1893–1898
11. Moosmann F, Stiller C (2011) Velodyne SLAM. In: Proceedings of IEEE Intelligent Vehicles Symposium, pp 393–398
12. Tokorodani K, Hashimoto M, Aihara Y, Takahashi K (2019) Point-cloud mapping using lidar mounted on two-wheeled vehicle based on NDT scan matching. In: Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, pp 446–452
13. Milstein A (2008) Occupancy grid maps for localization and mapping. In: Xing-Jian J (ed) Motion planning. InTech, London, pp 382–408
14. Muro S, Matsui Y, Hashimoto M, Takahashi K (2019) Moving-object tracking with lidar mounted on two-wheeled vehicle. In: Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics, pp 453–459
15. Tanaka S, Koshiro C, Yamaji M, Hashimoto M, Takahashi K (2020) Point cloud mapping and merging in GNSS-denied and dynamic environments using only onboard scanning LiDAR. *Int J Adv Syst Meas* 13:275–288
16. Biber P, Strasser W (2003) The normal distributions transform: a new approach to laser scan matching. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems pp 2743–2748
17. Hashimoto M, Ogata S, Oba F, Murayama T (2006) A laser based multi-target tracking for mobile robot. *Intell Auton Syst* 9:135–144
18. Tamura Y, Murabayashi R, Hashimoto M, Takahashi K (2017) Hierarchical cooperative tracking of vehicles and people using laser scanners mounted on multiple mobile robots. *Int J Adv Intell Syst* 10:90–101
19. Rusu RB, Cousins S (2011), 3D is here: point cloud library (PCL). In: Proceedings of 2011 IEEE International Conference on Robotics and Automation

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.