LiDAR-Based Simultaneous Localization, Mapping, and Multiobject Tracking: Combining Semantic Segmentation and Interacting Multiple Model-Extended Object Tracking

Akio Okumura
Graduate School of Science and Engineering
Doshisha University
Kyotanabe, Kyoto, Japan
e-mail: ctwn0155@mail4.doshisha.ac.jp

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

***Abstract*—This paper proposes a method for Simultaneous Localization And Mapping (SLAM) and MultiObject Tracking (MOT) using an onboard Light Detection And Ranging (LiDAR) system in dynamic road environments. LiDAR-based SLAM involves Normal Distributions Transform (NDT)-graph SLAM, and MOT combines an Interacting Multiple Model estimator with Extended Object Tracking (IMM-EOT). To effectively handle dynamic scenes, semantic segmentation using RangeNet++ classifies LiDAR measurements into those related to stationary or moving objects. NDT-graph SLAM uses the measurements related to stationary objects for robust mapping and localization, whereas IMM-EOT uses the measurements related to moving and potentially moving objects to simultaneously estimate the motions and shapes of objects exhibiting various motion patterns. The effectiveness of the proposed method is demonstrated through experiments on the SemanticKITTI and KITTI tracking datasets.**

Keywords*— *LiDAR; NDT-graph SLAM; multiobject tracking; semantic segmentation; IMM extended object tracking.

I. INTRODUCTION

Many Advanced Driver Assistance Systems (ADAS) and Autonomous Driving Systems (ADS) have been developed for Intelligent Transportation Systems (ITS) [1][2]. Simultaneous Localization And Mapping (SLAM) and MultiObject Tracking (MOT) using onboard sensors—such as cameras, Light Detection And Ranging (LiDAR) sensors, and radars—are key technologies that support ADAS and ADS [3][4]. This study focuses on LiDAR-based SLAM and MOT (SLAMMOT).

Typical LiDAR SLAM methods rely on scan-matching-based SLAM, which employs the iterative closest-point algorithm and the Normal Distributions Transform (NDT) [5]. However, the accuracy of scan-matching-based LiDAR SLAM degrades in dynamic environments with many coexisting moving objects, such as cars and two-wheelers.

Traditionally, LiDAR measurements related to moving objects were removed using the occupancy-grid method [6], which classifies LiDAR measurements into stationary objects (hereinafter "stationary LiDAR measurements") and moving objects (hereinafter "moving LiDAR measurements") and uses them for SLAM and MOT, respectively. However, occupancy-grid-based classification often fails to discriminate among different types of moving objects and frequently misdetects potentially moving objects—such as parked vehicles or vehicles stopping at red lights—as stationary objects. These limitations degrade the performance of ADAS and ADS.

Graph SLAM [7] is a popular method used for avoiding errors that accumulate over the travel time of the ego vehicle in scan-matching-based SLAM. A key component of graph SLAM is loop detection, which identifies previously visited locations. Many loop-detection methods have been proposed [5].

Conventional loop-detection methods rely on geometric point features, such as Fast Point Feature Histograms (FPFH) [8] or geometric surface-shape features (e.g., plains and poles) [9]. However, these geometric features cannot reliably distinguish objects with similar shapes; for example, traffic signs, utility poles, and trees are often confused, leading to false loop detection.

Recent studies have sought deep-learning-based object-recognition methods [10]. For example, the semantic segmentation method enhances the accuracy and robustness of LiDAR SLAM, and many semantic SLAM methods have been proposed [11][12][13]. Nevertheless, effectively integrating semantic information into LiDAR SLAM remains challenging.

Conventional MOT methods typically follow a two-step estimation sequence. First, object shapes are estimated as bounding boxes based on LiDAR measurements. Then, object motion parameters, such as the position and velocity, are estimated from the center position of the bounding box using Bayesian filters, including Kalman and particle filters [12] [14]. In this approach, the accuracy of motion parameter estimates strongly depends on the accuracy of shape estimation, and uncertainties (error covariances) in motion parameter and shape estimates are inaccurately calculated. Furthermore, because most conventional MOT methods assume nearly constant object velocities [11], their tracking performance degrades during sudden motion changes, such as sudden starts or stops. These limitations reduce the reliability of MOT in ADAS and ADS.

Extended Object Tracking (EOT) methods [15][16] resolve these limitations by simultaneously estimating motion parameters and shapes of objects, along with their uncertainties. The Interacting Multiple Model (IMM) framework has been combined with EOT (IMM-EOT) to accurately track objects exhibiting various behaviors, including stopping, near-constant-velocity motion, and sudden accelerations or decelerations, in the ITS domain [17][18].

Hence, this paper presents a SLAMMOT method for vehicle-mounted LiDAR systems in dynamic road environments. Extending our previous works [13][18], the proposed method integrates semantic segmentation and IMM-EOT.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

In [13], semantic information was used only for loop detection in LiDAR SLAM, and the moving LiDAR measurements were removed using occupancy-grid method. This approach retained potentially moving LiDAR measurements of parked or temporarily stopped vehicles in the environment maps. This paper applies semantic segmentation to remove both moving and potentially moving LiDAR measurements, thereby improving SLAM performance.

Furthermore, the IMM-EOT-based object tracking method reported in our previous work [18] utilized roadside LiDAR. When the LiDAR is fixed at an elevated position (e.g., at the top of a traffic-light pole), objects can be easily detected using background subtraction. Rather than evaluating the tracking performance through simulation experiments alone [18], this paper evaluates the newly proposed method on real-world open datasets. Using vehicle-mounted LiDAR, our MOT scheme integrates semantic segmentation and IMM-EOT. To the best of our knowledge, these have not been simultaneously implemented in LiDAR MOT.

The structure of this article is as follows: Section II overviews the LiDAR SLAMMOT approach, and Sections III and IV explain loop detection and MOT methods, respectively. Section V presents the experimental evaluation using the SemanticKITTI and KITTI tracking datasets [19][20]. The paper concludes with Section VI.

II. OVERVIEW OF LIDAR SLAMMOT

Figure 1 shows a sequence of our developed LiDAR SLAMMOT system. Semantic segmentation is performed based on the point-cloud segmentation model RangeNet++ [21]. LiDAR SLAM combines NDT SLAM with graph SLAM (NDT-graph SLAM), whereas MOT employs IMM-EOT, which integrates the Multiplicative Error Model-Extended Kalman Filter (MEM-EKF) framework [16] with an IMM estimator (IMM-MEM-EKF) [18].

RangeNet++ was selected because it requires less memory and computational cost than voxel-based and point-based semantic segmentation methods [22][23]. Among several EOT methods, including random matrix and random hypersurface model-based methods [15], we selected the MEM-EKF [16] because it estimates the object motion and shape states, along with their uncertainties (error covariances), within the Kalman filter framework.

LiDAR measurements are typically captured by scanning laser beams over the target area. Vehicle movement introduces motion artifacts (distortions) in the LiDAR measurements, degrading the SLAMMOT accuracy. Such motion artifacts are corrected with an unscented Kalman filter-based algorithm [24]. The corrected LiDAR measurements are then processed by RangeNet++ to obtain the semantic information. We consider 12 semantic classes: people, cars, two-wheelers, other vehicles, buildings, roads, fences, vegetation, lawns, trunks, traffic signs, and poles. After mapping the LiDAR measurements with their semantic labels onto a voxel map, the semantic class of each voxel is determined by majority voting of the semantic classes of the measurements.

LiDAR measurements related to moving objects, such as cars, two-wheelers, and people, which degrade the accuracy of NDT SLAM, are removed based on their semantic labels, leaving only the stationary LiDAR measurements for SLAM.

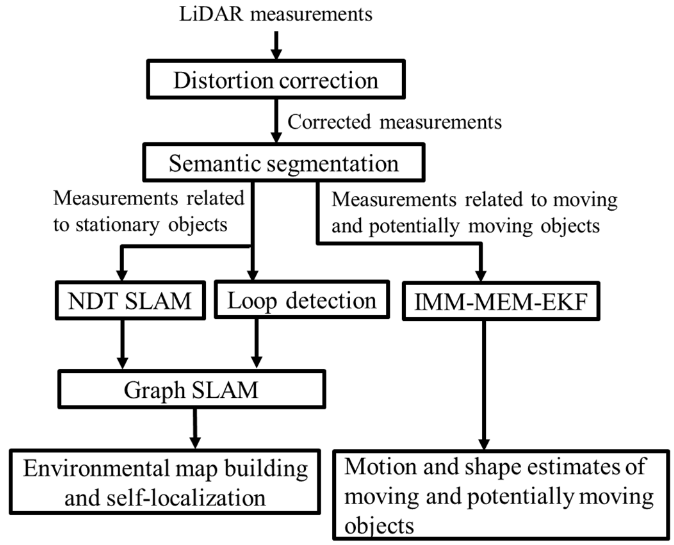


Figure 1. Sequence of LiDAR SLAMMOT.

The accuracy of NDT SLAM also deteriorates over time as errors accumulate. This error is mitigated by the loop-detection method within the graph SLAM framework [7]. Moving and potentially moving LiDAR measurements detected through the semantic segmentation method are applied to object tracking using the IMM-MEM-EKF method.

III. LOOP DETECTION AND RELATIVE POSE ESTIMATE

A. Loop Detection

Detection of loops (revisited locations) is a crucial problem in graph SLAM. Loops are selected from a set of loop candidates. To identify loop candidates, we define two feature descriptors, $F = (f_1, f_2, \ldots, f_6)$ and $G = (g_1, g_2, \ldots, g_6)$. Here, F and G are obtained from LiDAR measurements captured at the old and current locations of a vehicle, respectively, and f_i and g_i ($i = 1, 2, \ldots, 6$) denote the numbers of objects belonging to the i-th semantic class of stationary objects (buildings, fences, poles, traffic signs, vegetation, and trunks). Loop candidates are then identified based on the following similarity indicator:

$$\text{Similarity indicator} = \frac{\sum_{i=1}^{6}\{\max(f_i, g_i) - |f_i - g_i|\}}{\sum_{i=1}^{6}\max(f_i, g_i)} \tag{1}$$

When the similarity indicator is high, the old locations of the vehicle are identified as loop candidates.

Next, loops are selected from the identified loop candidates. NDT SLAM estimates the relative pose of the vehicle from the stationary LiDAR measurements captured at each loop candidate and at the current location. The overlap indicator between the two sets of stationary LiDAR measurements is then calculated as

$$\text{Overlap indicator} = \frac{1}{n}\sum_{i=1}^{n}\text{overlap}(d_i) \tag{2}$$

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

where n denotes the number of stationary LiDAR measurements captured at the current location, and d_i denotes the nearest-neighbor distance between the measurements at the current location and those at each loop candidate. If d_i is shorter than 0.1 m, the overlap function (d_i) is set to one, otherwise, it is set to zero.

Finally, loop candidates with high overlap-indicator values are accepted as loops.

B. Relative Pose Estimate

The relative poses of the vehicle at the current location and at loops are calculated by NDT scan matching and provided as loop constraints in graph SLAM. Because NDT scan matching is a local registration method, an accurate initial estimation of the relative pose (i.e., coarse registration) is required to avoid local minima problems. In this paper, coarse registration is performed based on Semantic and Geometric point Feature Histograms (SGFH) [13].

First, the stationary LiDAR measurements captured at the current vehicle location are mapped onto a voxel grid map and downsampled using a voxel grid filter. As shown in Figure 2, the centroid (called the feature point) of the i-th voxel is denoted as A_i and the feature point of the surrounding j-th voxel (j = 1, 2, ..., n) is denoted as A_j. After calculating the geometric three-angle features α_j, β_j, and γ_j at each feature point, a geometric point feature histogram (33 dimensions in this study) is obtained based on FPFH [8].

A semantic feature histogram is also constructed using the semantic labels of the six stationary semantic classes: buildings, fences, poles, traffic signs, vegetation, and trunks. Semantic labels are assigned to the feature points A_i and A_j (i, j = 1, 2, ..., n). The six-dimensional Semantic Feature Histogram (SFH) related to A_i is then given by

$$\boldsymbol{SFH}(A_i) = \boldsymbol{SH}(A_i) + \frac{1}{n}\sum_{j=1}^{n} w_j \boldsymbol{SH}(A_j) \qquad (3)$$

where $\boldsymbol{SH}(A_i)$ and $\boldsymbol{SH}(A_j)$ denote the SFHs at the feature points A_i and A_j, respectively. The weight w_i is given as the inverse of the distance between A_i and A_j.

Finally, the six-dimensional SFH is combined with the 33-dimensional geometric feature histogram to form the 39-dimensional SGFH descriptor. The same procedure is applied to the stationary LiDAR measurements captured at each loop to obtain the feature points (B_i and B_j) and SGFH descriptors.

Feature points A_i and B_i are then matched using a random sample consensus-based method [25], completing the coarse registration. The relative pose between the current and loop locations, refined by NDT scan matching, is subsequently used as a loop constraint in graph SLAM to correct the accumulated error in NDT SLAM.

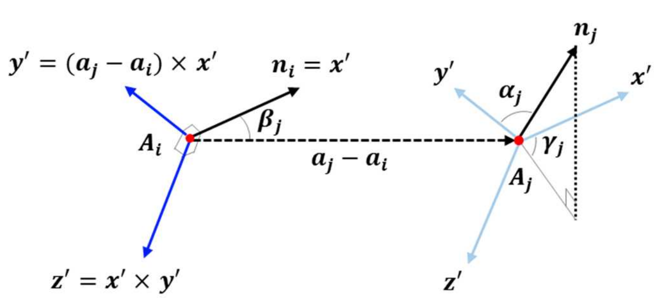


Figure 2. Feature points and angle features in FPFH.

IV. Multi-Object Tracking

A. Motion, Shape, and Measurement Models

The moving LiDAR measurements extracted using the semantic segmentation method are clustered via density-based spatial clustering of applications with noise [26]. To track moving objects, the clustered moving LiDAR measurements are processed using the IMM-MEM-EKF method. To accommodate diverse object motions in real road environments, the following motion modes of target objects are considered:

- Mode 1: Constant-position mode for temporarily stationary objects.
- Mode 2: Constant-velocity mode for an object moving at a nearly constant linear or turning velocity.
- Mode 3: Acceleration mode for objects that suddenly accelerate or decelerate.

The motion state in mode m (m = 1, 2, 3) is modeled by the following state equation [18]:

$$\boldsymbol{r}_{m,t} = \boldsymbol{f}_m(\boldsymbol{r}_{m,t-1}, \Delta\boldsymbol{r}_{m,t-1}) \qquad (4)$$

where t and $t-1$ denote the time steps. $\boldsymbol{r}_m$ denotes the motion-state variable in mode m, and $\Delta\boldsymbol{r}_m$ is the related disturbance with covariance matrix $\boldsymbol{C}_m^{\Delta r}$. $\boldsymbol{f}_m$ is a function of $\boldsymbol{r}_m$ and $\Delta\boldsymbol{r}_m$.

The motion-state variables for modes 1, 2, and 3 are $\boldsymbol{r}_1 = (x, y)^T$, $\boldsymbol{r}_2 = (x, y, \theta, v, \omega)^T$, and $\boldsymbol{r}_3 = (x, y, \theta, v, \dot{v})^T$, respectively, where ($x$, y) and θ denote the position and moving direction of an object, respectively, in the world coordinate system (Figure 3). Further, v and ω denote the linear and turning velocities, respectively, and $\dot{v}$ denotes the linear acceleration.

The shape state of an object is modeled as follows:

$$\boldsymbol{p}_t = \boldsymbol{p}_{t-1} + \Delta\boldsymbol{p}_{t-1} \qquad (5)$$

where $\boldsymbol{p} = (\phi, l_1, l_2)^T$ denotes the shape-state variable, consisting of the heading angle ϕ and object size (l_1, l_2). $\Delta\boldsymbol{p}$ is the related disturbance with covariance matrix $\boldsymbol{C}^{\Delta p}$.

Denoting N_t as the number of moving LiDAR measurements of an object at time t (t = 0, 1, 2, ...) and $\boldsymbol{y}_t^{(i)} = (y_{1t}^{(i)}, y_{2t}^{(i)})^T$ as the i-th measurement (i = 1, 2, ... , N_t) in the

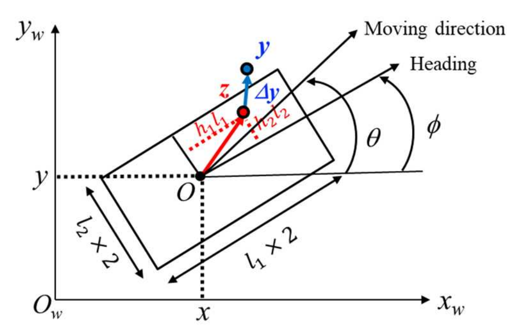


Figure 3. Motion and shape-state variables (top view).

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

world coordinate system (Figure 3), the measurement model is given by

$$\boldsymbol{y}_t^{(i)} = \boldsymbol{H}_m \boldsymbol{r}_{m,t} + \boldsymbol{S}_t \boldsymbol{h}_t^{(i)} + \Delta \boldsymbol{y}_t^{(i)} \quad (6)$$

where $\boldsymbol{H}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\boldsymbol{H}_2 = \boldsymbol{H}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$, and $\boldsymbol{S}_t = \begin{pmatrix} S_{1t} \\ S_{2t} \end{pmatrix} = \begin{pmatrix} \cos\phi_t & -\sin\phi_t \\ \sin\phi_t & \cos\phi_t \end{pmatrix} \begin{pmatrix} l_{1t} & 0 \\ 0 & l_{2t} \end{pmatrix}$.

$\boldsymbol{h}_t^{(i)} = (h_1^{(i)}, h_2^{(i)})^T$ are uniformly distributed random variables in the interval $-1 \le h_1^{(i)} \le 1$ and $-1 \le h_2^{(i)} \le 1$. The variable $\boldsymbol{h}_t^{(i)}$ is assumed as Gaussian white noise with zero mean and covariance $\boldsymbol{C}^h = \boldsymbol{E}_2 / 3$, where $\boldsymbol{E}_2$ is the two-dimensional identity matrix. $\Delta \boldsymbol{y}_t^{(i)}$ denotes the measurement noise with related covariance matrix $\boldsymbol{C}^{\Delta y}$.

B. Motion Estimation of Object

Let $\hat{\mu}_{m,t-1}^{(N_{t-1})}$ be the probability of mode m (m = 1, 2, 3) occurring at time $t-1$. The conditional state estimate and its related error covariance for mode m are denoted by $\hat{\boldsymbol{r}}_{m,t-1}^{(N_{t-1})}$ and $\boldsymbol{C}_{m,t-1}^{r(N_{t-1})}$, respectively. These quantities are mixed as

$$\left.\begin{aligned} \overline{\boldsymbol{r}}_{m,t-1}^{(N_{t-1})} &= \sum_{n=1}^{3} \alpha_{mn} \hat{\boldsymbol{r}}_{n,t-1}^{(N_{t-1})} \\ \overline{\boldsymbol{C}}_{m,t-1}^{r(N_{t-1})} &= \sum_{n=1}^{3} \alpha_{mn} [\boldsymbol{C}_{n,t-1}^{r(N_{t-1})} + (\hat{\boldsymbol{r}}_{n,t-1}^{(N_{t-1})} - \overline{\boldsymbol{r}}_{m,t-1}^{(N_{t-1})})(\bullet)^T] \end{aligned}\right\} \quad (7)$$

where N_{t-1} is the number of moving LiDAR measurements obtained at time $t-1$. The mixing weight α_{mn} (m, n = 1, 2, 3) is given by $\alpha_{mn} = T_{mn} \hat{\mu}_{m,t-1}^{(N_{t-1})} / \sum_{m=1}^{3} T_{mn} \hat{\mu}_{m,t-1}^{(N_{t-1})}$. T_{mn} is the transition probability from mode m to n. In this study, the transition probabilities are set to T_{mn} = 0.9 for $m = n$ and T_{mn} = 0.05 for $m \neq n$.

Using the EKF, the prediction and its related error covariance for mode m at time t are computed as follows:

$$\left.\begin{aligned} \hat{\boldsymbol{r}}_{m,t}^{(0)} &= \boldsymbol{f}_m (\overline{\boldsymbol{r}}_{m,t-1}^{(N_{t-1})}) \\ \boldsymbol{C}_{m,t}^{r(0)} &= \nabla \boldsymbol{F}_{m,t-1} \overline{\boldsymbol{C}}_{m,t-1}^{r(N_{t-1})} \nabla \boldsymbol{F}_{m,t-1}^T + \nabla \boldsymbol{G}_{m,t-1} \boldsymbol{C}_w^r \nabla \boldsymbol{G}_{m,t-1}^T \end{aligned}\right\} \quad (8)$$

where $\nabla \boldsymbol{F}$ and $\nabla \boldsymbol{G}$ are the Jacobians of the motion model $\boldsymbol{f}_m$ pertaining to the state $\overline{\boldsymbol{x}}_{m,t-1}$ and noise $\Delta \boldsymbol{w}_{m,t-1}$, respectively.

When the i-th moving LiDAR measurement ($i = 1, 2, \cdots, N_t$) is obtained at time t, the motion estimate $\hat{\boldsymbol{r}}_{m,t}^{(i)}$, its related error covariance $\boldsymbol{C}_{m,t}^{r(i)}$, and the likelihood $\lambda_{m,t}^{(i)}$ of mode m are updated as follows:

$$\left.\begin{aligned} \hat{\boldsymbol{r}}_{m,t}^{(i)} &= \hat{\boldsymbol{r}}_{m,t}^{(i-1)} + \boldsymbol{C}_{m,t}^{ry(i)} (\boldsymbol{C}_{m,t}^{y(i)})^{-1} \tilde{\boldsymbol{y}}_{m,t}^{(i)} \\ \boldsymbol{C}_{m,t}^{r(i)} &= \boldsymbol{C}_{m,t}^{r(i-1)} - \boldsymbol{C}_{m,t}^{ry(i)} (\boldsymbol{C}_{m,t}^{y(i)})^{-1} (\boldsymbol{C}_{m,t}^{ry(i)})^T \end{aligned}\right\} \quad (9)$$

$$\lambda_{m,t}^{(i)} = \frac{1}{2\pi \sqrt{\left|\boldsymbol{C}_{m,t}^{y(i)}\right|}} \exp[-\frac{1}{2} (\tilde{\boldsymbol{y}}_{m,t}^{(i)})^T (\boldsymbol{C}_{m,t}^{y(i)})^{-1} \tilde{\boldsymbol{y}}_{m,t}^{(i)}] \quad (10)$$

where $\tilde{\boldsymbol{y}}_{m,t}^{(i)}$ and $\boldsymbol{C}_{m,t}^{y(i)}$ denote the error in the predicted measurement and its related covariance, respectively:

$$\left.\begin{aligned} \tilde{\boldsymbol{y}}_{m,t}^{(i)} &= \boldsymbol{y}_t^{(i)} - \boldsymbol{H}_m \hat{\boldsymbol{r}}_{m,t}^{(i-1)} \\ \boldsymbol{C}_{m,t}^{y(i)} &= \boldsymbol{H}_m \boldsymbol{C}_{m,t}^{r(i-1)} \boldsymbol{H}_m^T + \hat{\boldsymbol{S}}_t^{(i-1)} \boldsymbol{C}^h (\hat{\boldsymbol{S}}_t^{(i-1)})^T \\ &\quad + \mathrm{trace}\{\boldsymbol{C}_t^{p(i-1)} (\hat{\boldsymbol{J}}_{bt}^{(i-1)})^T \boldsymbol{C}^h (\hat{\boldsymbol{J}}_{at}^{(i-1)})\} + \boldsymbol{C}_m^{\Delta r} \end{aligned}\right\} \quad (11)$$

where $\hat{\boldsymbol{J}}_{at}^{(i-1)}$ and $\hat{\boldsymbol{J}}_{bt}^{(i-1)}$ (a, b =1, 2) are the Jacobians of S_{1t} and S_{2t}, respectively.

Equations (9) and (11) follow the derivations in [16]. In terms of likelihood $\lambda_{m,t}^{(i)}$, the mode probability for mode n is updated as

$$\hat{\mu}_{n,t}^{(N_t)} = \frac{\overline{\mu}_{n,t}^{(N_{t-1})} \Lambda_{n,t}}{\sum_{m=1}^{3} \overline{\mu}_{m,t}^{(N_{t-1})} \Lambda_{m,t}} \quad (12)$$

where $\Lambda_{m,t} = \prod_{i=1}^{N_t} \lambda_{m,t}^{(i)}$ and $\overline{\mu}_{n,t}^{(N_{t-1})} = \sum_{m=1}^{3} T_{mn} \hat{\mu}_{m,t-1}^{(N_{t-1})}$

Finally, the overall motion estimate and its related error covariance are obtained as

$$\left.\begin{aligned} \hat{\boldsymbol{r}}_t^{(N_t)} &= \sum_{m=1}^{3} \hat{\mu}_{m,t}^{(N_t)} \hat{\boldsymbol{r}}_{m,t}^{(N_t)} \\ \boldsymbol{C}_t^{r(N_t)} &= \sum_{m=1}^{3} \hat{\mu}_{m,t}^{(N_t)} [\boldsymbol{C}_{m,t}^{r(N_t)} + (\hat{\boldsymbol{r}}_{m,t}^{(N_t)} - \hat{\boldsymbol{r}}_t^{(N_t)})(\bullet)^T] \end{aligned}\right\} \quad (13)$$

Tracking in multiobject environments requires data association (one-to-one matching between tracked objects and clustered moving LiDAR measurements). To this end, the object shape (assumed rectangular) is extracted from the moving LiDAR measurements using the rotating-calipers method [27]. The Intersection over Union (IoU) between the extracted rectangle and the rectangle estimated by the IMM-MEM-EKF method is then calculated. Based on the IoU, data association is performed using a simple online real-time tracking algorithm [28].

C. Shape Estimation of Object

The predicted shape of the object and its related error covariance at time t are given by

$$\left.\begin{aligned} \hat{\boldsymbol{p}}_t^{(0)} &= \hat{\boldsymbol{p}}_{t-1}^{(N_{t-1})} + \boldsymbol{A} \hat{\boldsymbol{r}}_{t-1}^{(N_{t-1})} \\ \boldsymbol{C}_t^{p(0)} &= \boldsymbol{C}_{t-1}^{p(N_{t-1})} + \boldsymbol{C}^{\Delta p} + \boldsymbol{A} \boldsymbol{C}_{t-1}^{r(N_{t-1})} \boldsymbol{A}^T \end{aligned}\right\} \quad (14)$$

where the matrix $\boldsymbol{A}$ is used for extracting the rotational velocity $\hat{\omega}_{t-1}^{(N_{t-1})}$.

Equation (5) assumes that the object maintains a constant heading angle. The final term in (14) incorporates the correlation between the heading angle and rotational velocity.

For the i-th moving LiDAR measurement (i = 1, 2, … , N_t) obtained at time t, the shape estimate and its related error covariance are computed as follows:

$$\left.\begin{aligned} \hat{\boldsymbol{p}}_t^{(i)} &= \hat{\boldsymbol{p}}_t^{(i-1)} + \boldsymbol{C}_t^{pY(i)} (\boldsymbol{C}_t^{Y(i)})^{-1} (\boldsymbol{Y}_t^{(i)} - \overline{\boldsymbol{Y}}_t^{(i)}) \\ \boldsymbol{C}_t^{p(i)} &= \boldsymbol{C}_t^{p(i-1)} - \boldsymbol{C}_t^{pY(i)} (\boldsymbol{C}_t^{Y(i)})^{-1} (\boldsymbol{C}_t^{pY(i)})^T \end{aligned}\right\} \quad (15)$$

where $\boldsymbol{Y}_t^{(i)}$ is the pseudo-measurement of $\boldsymbol{y}_t^{(i)}$. $\overline{\boldsymbol{Y}}_t^{(i)}$ and $\boldsymbol{C}_t^{Y(i)}$ are the predicted pseudo-measurement and its related error covariance, respectively, and $\boldsymbol{C}_t^{pY(i)}$ represents the correlation

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

between the pseudo-measurement and the motion estimate. These quantities are defined in [16].

V. Classification of LiDAR Measurements

A. Dataset

The performance of the proposed SLAMMOT approach is evaluated using the SemanticKITTI and KITTI tracking datasets [19][20]. The SemanticKITTI dataset contains sequences in which the ego vehicle travels along routes with many loops; however, the ground-truth positions and sizes of the target objects are not provided. Conversely, the KITTI tracking dataset presents ground-truth annotations for object positions and sizes but does not include loop-rich routes suitable for SLAM evaluation. For this reason, the SLAM and MOT are evaluated on the SemanticKITTI and KITTI tracking datasets, respectively.

In both datasets, the ego vehicle is equipped with a 64-layer LiDAR (Velodyne HDL-64E). The horizontal and vertical fields of view of the LiDAR are 360° and 26.8°, respectively. During each LiDAR sampling period (0.1 s), the 64 laser beams complete one full rotation in the horizontal direction, producing approximately 120,000 measurements.

B. Object Recognition Result

First, the object-recognition performance of the SLAMMOT approach is evaluated. Because the LiDAR-equipped ego vehicles in both datasets travel in similar environments, RangeNet++ is trained on the SemanticKITTI dataset alone. Table I shows the result of object recognition. As shown in the table, large objects—such as cars, roads, buildings, and vegetation—are accurately detected, whereas the classification accuracy is lower for small objects, including people and traffic signs.

For the six semantic voxel classes used in the loop-detection method (buildings, fences, poles, traffic signs, trunks, and vegetation), the average IoU, precision, and recall are computed as 54.8 %, 73.5 %, and 67.0 %, respectively. For the four semantic voxel classes used in MOT (cars, two-wheelers, other vehicles, and people), the average IoU, precision, and recall are 50.6 %, 70.8 %, and 60.1 %, respectively.

C. SLAM Result

Among the 11 sequences in the SemanticKITTI dataset, sequences 00, 02, 05, and 08, which contain many loops, are selected for evaluating the SLAM performance. The SLAM evaluation is conducted under the following cases.

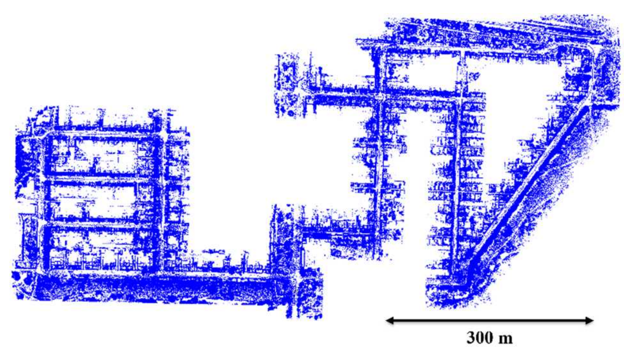
- Case 1 (proposed method): SLAM using only the stationary LiDAR measurements extracted by semantic segmentation. The moving and potentially moving LiDAR measurements are removed.
- Case 2 (our previous method [13]): SLAM using the stationary LiDAR measurements extracted by the occupancy-grid method. The moving LiDAR measurements are removed, but the potentially moving LiDAR measurements are retained as stationary.
- Case 3: SLAM using both the stationary and moving LiDAR measurements without removal.

Figures 4 (a) and (b) show the true environmental map and corresponding mapping result, respectively, for sequence 08 in case 1. Figure 5 shows the SLAM-estimated trajectory of the ego vehicle in cases 1, 2, and 3. The ego-vehicle pose is

TABLE I. Performance of Object Classification

Class	IoU [%]	Precision [%]	Recall [%]
Car	89.2	91.4	97.4
Two-wheeler	45.4	66.2	59.2
Other vehicles	46.6	70.5	57.8
Person	21.4	54.9	26.0
Road	93.1	97.0	96.9
Lawn	68.7	88.3	75.5
Building	80.9	88.4	90.5
Fence	46.7	67.5	60.2
Vegetation	81.4	86.8	93.0
Trunk	47.0	67.9	60.4
Pole	43.9	57.1	65.6
Traffic sign	28.9	73.3	32.2

(a) True map.



(b) Built map (case 1).

Figure 4. Environment mapping result for sequence 08 (top view).

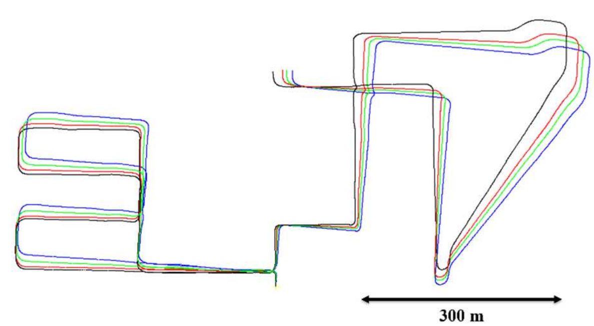


Figure 5. Estimated trajectory of the ego vehicle in sequence 08 (top view). The black line indicates the ground truth. The red, blue, and green lines indicate the results for cases 1, 2, and 3, respectively.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

accurately estimated in case 1, whereas case 3 exhibits a large localization error.

The SLAM performance is quantitatively evaluated using Absolute Trajectory Error (ATE), defined as the error in the ego vehicle's trajectory. Table II shows the result. On all four sequences, case 1 is the lowest ATE among the three cases.

D. MOT Result

The MOT performance is evaluated on the KITTI tracking dataset. As an example, Figure 6 shows the MOT result of sequence 02. The black dotted line indicates the trajectory of the ego vehicle. The rectangles indicate the estimated bounding boxes of objects, and the lines indicate the heading direction of moving objects. The different colored dots indicate the moving LiDAR measurements extracted by semantic segmentation. The MOT performance is quantitatively evaluated using IoU metric, which measures the overlap between the true and estimated bounding boxes of moving objects. The KITTI tracking dataset provides ground-truth bounding boxes of objects visible to the camera mounted on the ego vehicle. For comparison, moving objects are also tracked using the conventional single model-based MEM-EKF.

Table III shows the results for five sequences. The proposed multimodel-based MEM-EKF outperforms the single model-based MEM-EKF for sequences 02, 03, and 06, which represent intersection environments where car motion frequently changes. For sequence 05, which represents a straight-road environment, both frameworks exhibit low MOT accuracy because most cars were located ahead of the ego vehicle, and the LiDAR measurements were acquired from the rear ends of the cars. Both frameworks achieve similar MOT accuracies for sequence 08, which represents a highway environment where most cars travel at nearly constant velocity. Overall, multimodel-based MEM-EKF achieves higher MOT accuracy than the single model-based MEM-EKF in scenes with frequent transitions among motion, such as those induced by intersections.

TABLE II. ATE IN SLAM

Sequence	Case 1	Case 2	Case 3
00	**10.36** m	10.74 m	19.02 m
02	**55.46** m	61.42 m	108.99 m
05	**8.25** m	8.69 m	10.54 m
08	**12.04** m	15.10 m	27.44 m

The smallest ATEs are indicated in bold.

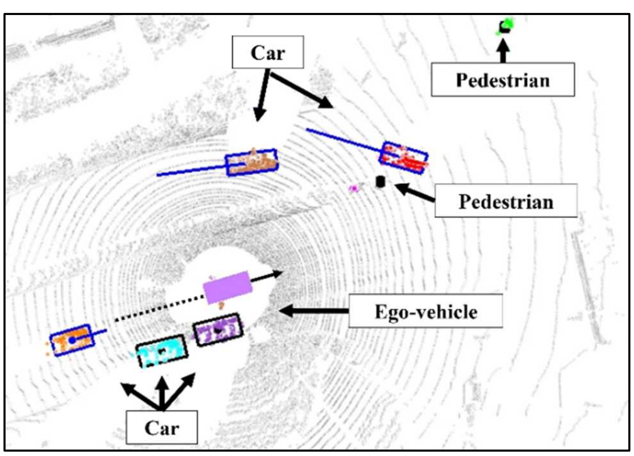


Figure 6. MOT result for sequence 02 of the KITTI tracking dataset (top view).

TABLE III. IOU PERFORMANCE OF MOT

Sequence	Multimodel-based EOT	Single model-based EOT
02	**71.3** %	67.6 %
03	**80.7** %	75.1 %
05	45.0 %	**51.3** %
06	**73.1** %	70.7 %
08	62.6 %	**64.8** %

The highest IoUs are indicated in bold.

VI. CONCLUSION AND FUTURE WORK

This paper presented a LiDAR SLAMMOT approach based on the semantic segmentation and IMM-EOT methods. After discriminating stationary and moving objects using RangeNet++ based on onboard LiDAR measurements, the stationary LiDAR measurements were input to NDT-Graph SLAM, while the moving and potentially moving LiDAR measurements were used for MOT using the IMM-MEM-EKF. The performance of the proposed method was evaluated on the SemanticKITTI and KITTI tracking datasets.

Further evaluation and improvement of the proposed method are required, including MOT for small target objects, such as pedestrians and two-wheelers, as well as the development of a tightly coupled SLAM and MOT system. In this paper, SLAMMOT was executed offline on a laptop computer. As future work, we aim to reduce the computational time by optimizing the program code and employing embedded hardware for real-time operation.

ACKNOWLEDGMENT

This study was partially supported by the KAKENHI Grant #23K03781, the Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] J. Zhao et al., "Autonomous Driving System: A Comprehensive Survey," Expert Systems with Applications 242, pp. 1–27, 2023.

[2] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A Survey of Driving Safety with Sensing, Vehicular Communications, and Artificial Intelligence-based Collision Avoidance," IEEE Trans. Intelligent Transportation Systems, vol. 23, pp. 6142–6163, 2022.

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

[4] E. Arnold et al., "A Survey on 3D Object Detection Methods for Autonomous Driving Applications," IEEE Trans. Intelligent Transportation Systems, vol. 20, pp. 3782–3795, 2019.

[5] Y. Zhang, P. Shi, and J. Li, "3D LiDAR SLAM: A Survey," The Photogrammetric Record, vol. 39, pp. 457–517, 2024.

[6] S. Steyer, G. Tanzmeister, and D. Wollherr, "Object Tracking Based on Evidential Dynamic Occupancy Grids in Urban Environments," Proc. 2017 IEEE Intelligent Vehicles Symposiums, pp. 1064–1070, 2017.

[7] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-based SLAM," IEEE Intelligent Transportation Systems Magazine, vol. 2, pp. 31–43, 2010.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

[8] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D Registration,” Proc. IEEE Int. Conf. Robotics and Automation, pp. 3212–3217, 2009.

[9] 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.

[10] L. Wang and Y. Huang, “A Survey of 3D Point Cloud and Deep Learning-based Approaches for Scene Understanding in Autonomous Driving,” IEEE Intelligent Transportation Systems Magazine, vol. 14, pp. 135–154, 2022.

[11] Y. Wang, Z. Wang, and X. Wu, “An Efficient and Accurate 3D SLAM Method for Dynamic Environment,” Proc. 4th Int. Conf. Robotics and Computer Vision, pp. 148–153, 2022.

[12] Z. Ying and H. Li, “IMM-SLAMMOT: Tightly-Coupled SLAM and IMM-based Multi-Object Tracking,” IEEE Trans. Intelligent Vehicles, vol. 9, pp. 3964–3974, 2024.

[13] S. Muramatsu, M. Hashimoto, and K. Takahashi, “Performance Improvement of Loop Closure Detection Using Semantic Segmentation in LiDAR SLAM,” Proc. the 10th Int. Conf. Advances in Sensors, Actuators, Metering and Sensing, pp. 45–51, 2025.

[14] Y. Tamura, R. Murabayashi, M. Hashimoto, and K. Takahashi, “Hierarchical Cooperative Tracking of Vehicles and People Using Laser Scanners Mounted on Multiple Mobile Robots,” Int. J. Advances in Intelligent Systems, vol. 10, pp. 90–101, 2017.

[15] K. Granstrom and M. Baum, “A Tutorial on Multiple Extended Object Tracking,” TechRxiv, pp. 1–36, 2022.

[16] S. Yang and M. Baum, “Tracking the Orientation and Axes Lengths of an Elliptical Extended Object,” IEEE Trans. Signal Process, vol. 67, pp. 4720–4729, 2019.

[17] S. Wang, C. Men, R. Li, and T. Yeo, “A Maneuvering Extended Target Tracking IMM Algorithm Based on Second-Order EKF,” IEEE Trans. Instrumentation and Measurements, vol. 73, pp. 1–11, 2024.

[18] Y. Nakatani, S. Uchiyama, M. Hashimoto, and K. Takahashi, “LiDAR-based Cooperative Tracking Using IMM Extended Object Tracking and Information Filter,” Proc. IEEE Region 10 Conf., pp. 51–55, 2025.

[19] J. Behley et al., “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” Proc. IEEE/CVF Int. Conf. Computer Vision, pp. 9297–9307, 2019.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3354–3361, 2012.

[21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," Proc. 2019 IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 4213-4220, 2019.

[22] A. Jhaldiyal and N. Chaudhary, “Semantic Segmentation of 3D LiDAR Data Using Deep Learning: A Review of Projection-Based Methods,” Applied Intelligence, vol. 53, pp. 6844–6855, 2023.

[23] H. Zhao, L. Jiang, J. Jia, P. H. K. Torr, and V. Koltun, "Point Transformer," Proc. IEEE/CVF Int. Conf. Computer Vision, pp. 16259–16268, 2021.

[24] R. Nakamura, I. Inaga, M. Hashimoto, and K. Takahashi, “SLAM-based Mapping Using Micromobility-Mounted Solid-State LiDAR,” Proc. the 11th IIAE Int. Conf. Intelligent Systems and Image Processing 2024, pp. 93–100, 2024.

[25] R. Nakamura, T. Kambe, M. Hashimoto, and K. Takahashi, “SLAM-based Mapping in Truck-and-Robot System for Last-Mile Delivery Automation,” Proc. the 2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications, pp. 31–37, 2023.

[26] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” Proc. the Second Int. Conf. Knowledge Discovery and Data Mining, pp. 226–231, 1996.

[27] G. Toussaint, “Solving Geometric Problems with the Rotating Calipers,” Proc. IEEE Mediterranean Electrotechnical Conf.'83, pp. 1-8, 1983.

[28] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple Online and Realtime Tracking,” Proc. IEEE Int. Conf. Image Process, pp. 3464–3468, 2016.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

Copyright (c) IARIA, 2026. ISBN: 978-1-68558-399-6