IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Semantic Localization Considering Uncertainty of Object Recognition

Naoki Akai[1], Takatsugu Hirayama[2], and Hiroshi Murase[1]

Abstract— **Semantics can be leveraged in ego-vehicle localization to improve robustness and accuracy because objects with the same labels can be correctly matched with each other. Object recognition has significantly improved owing to advances in machine learning algorithms. However, perfect object recognition is still challenging in real environments. Hence, the uncertainty of object recognition must be considered in localization. This paper proposes a novel localization method that integrates a supervised object recognition method, which predicts probabilistic distributions over object classes for individual sensor measurements, into the Bayesian network for localization. The proposed method uses the estimated probabilities and Dirichlet distribution to calculate the likelihood for estimating an ego-vehicle pose. Consequently, the uncertainty can be handled in localization. We present an implementation example of the proposed method using a particle filter and deep-neural-network-based point cloud semantic segmentation and evaluate it by simulation and the SemanticKITTI dataset. Experimental results show that the proposed method can accurately generate likelihood distribution even when object recognition accuracy is degraded, and its estimation accuracy is the highest compared to that of two conventional methods.**

I. Introduction

Object recognition has been significantly improved owing to advances in machine learning algorithms. In particular, deep neural network (DNN)-based semantic segmentation (SS) enables precise pixel- and laser-wise object recognition [1]–[4]. Ego-vehicle localization methods for leveraging semantics have been proposed [5]–[9]. The use of semantics enables the correct matching of objects having the same labels and improves localization robustness and accuracy. However, perfect object recognition is still challenging in real environments. Therefore, the uncertainty of object recognition must be considered in the localization. This paper proposes a novel localization method that integrates a supervised learning (SL)-based object recognition method into the Bayesian network for localization. In this work, we assume that the object recognition method estimates probabilistic distributions over object classes for each sensor measurement. Note that we define that using the probabilistic distributions, not a certain label, in the matching process is considering the uncertainty.

Figure 1 illustrates the graphical model of the proposed method. The vehicle pose, $\mathbf{x}$, is treated as a latent variable. The control input, $\mathbf{u}$, sensor measurement, $\mathbf{z}$, semantic map, $\mathbf{m}$, and hyperparameters for the SL-based object recognition, Θ, are treated as observable variables. The object recognition estimates probabilities over the object classes, $\mathbf{c}$, and we assume that it can be used as an observable variable. A training dataset for the object recognition, $D = \{Z, S\}$, is also treated as an observable variable, where Z and S are a set of sensor measurements and its annotation labels. The proposed method estimates posterior distribution over the current vehicle pose.

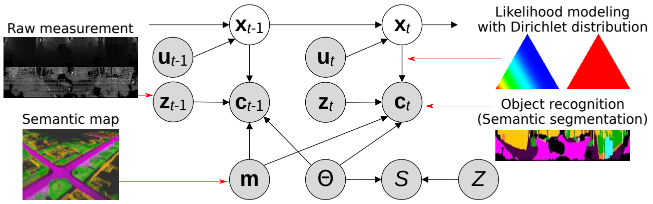

Fig. 1. Graphical model of the proposed method. The proposed method integrates a probabilistic object recognition method into the Bayesian network for localization.

The proposed method uses the object recognition results, i.e., prediction using the SL-based object recognition, as the observable variable, and it depends on the vehicle pose. Therefore, the prediction must be modeled to calculate the likelihood for estimating the vehicle pose. In this work, Dirichlet distribution is used to model the prediction because the prediction is discrete probabilistic distribution over the object classes. We refer to this model as the *class prediction model (CPM)*. Because Dirichlet distribution can represent a likelihood distribution over a discrete probabilistic distribution, the CPM enables to model possibility of misrecognition. As a result, the vehicle pose can be robustly and accurately estimated even when the SL-based object recognition is noisy.

We first derive the mathematical details of the proposed method and then present its implementation example using a particle filter (PF) and DNN-based point cloud SS. We validate the implemented method using simulation and the SemanticKITTI dataset [10]. In the simulation experiments, the robustness to degradation in the SS accuracy is validated, and it is shown that the proposed method can accurately generate the likelihood distribution even though the SS accuracy is considerably low. In the validation using the SemanticKITTI dataset, two other methods are compared with the proposed method and it is illustrated that the proposed method achieves the most accurate localization.

The contribution of this work is threefold.

- Proposing the novel localization method that integrates the SL-based object recognition and makes it possible to handle the object recognition uncertainty in the

[1]Naoki Akai and Hiroshi Murase are with the Graduate School of Informatics, Nagoya University, Nagoya 464-8603, Japan {akai, murase}@nagoya-u.jp

[2]Takatsugu Hirayama is with the Institute of Innovation for Future Society (MIRAI), Nagoya University, Nagoya 464-8601, Japan takatsugu.hirayama@nagoya-u.jp

localization

- Deriving the mathematical details of the proposed method and presenting the implementation example using the PF and DNN-based point cloud SS
- Showing robust and accurate localization performance from the experiments using the simulation and SemanticKITTI dataset.

The remainder of this paper is organized as follows. Section II summarizes related work. Section III presents the details of the proposed method. Section IV describes the implementation details of the proposed method. Section V and VI describe the experimental results. Section VII concludes this work.

II. Related work

Point cloud registration-based localization, e.g., [11], [12], is widely used in current autonomous navigation systems, e.g., [13], [14]. However, registration based solely on metric information sometimes causes obvious mismatches, e.g., vegetation is matched with a fence. To prevent these obvious mismatches, semantics of objects must be leveraged.

The 3D normal distributions transform (NDT) [15] is popular for autonomous driving, [16], [17], because it enables fast 3D point cloud registration. Zaganidis *et al.* [18] proposed semantic-assisted 3D NDT, where the object labels of the points are used to compute the cost function. The authors indicated that the method improved the accuracy, robustness, and speed of NDT registration, especially in unstructured environments. Zaganidis *et al.* [19] also presented the integration method of DNN-based SS into 3D point cloud registration including semantic-assisted generalized ICP (SE-GICP). SE-GICP is similar to the method presented in [20], where the labels are used to find more accurate correspondences. Chen *et al.* [8] used semantic ICP and achieved an accurate odometry estimate. These works leveraged semantics to improve point cloud registration; however, they did not consider the uncertainty of the object recognition.

Yi *et al.* [21] proposed the enhanced Markov localization method to support contextual representations of a robot's location. In the method, a monocular camera and a set of semantics are used for the spatial contexts of the object and robot, and the features extracted from the image are introduced to the Bayesian network for localization. These three variables are used to calculate the likelihood for estimating the robot pose and are modeled using Gaussian distributions; they are thereby referred to as the contextual measurement model. Atanasov *et al.* [22] presented the semantic observation model for monocular camera observations. In the model, the bearing measurement, class, and class recognition score are used as the observable variables. The authors also present an efficient method to solve the data association problem with semantics using a matrix permanent that was computed from the bipartite graph. Bowman *et al.* [23] extended the method and proposed the probabilistic data association method for semantic SLAM. In the method, semantics are introduced in the expectation-maximization (EM) procedure, which enables flexible and robust data association. These methods are similar to the proposed method because they handle semantics via probabilistic modeling. However, the use of Dirichlet distribution is not introduced in these methods.

Parkison *et al.* [24] presented semantic ICP through the EM algorithm. Semantic labels and point associations between two point clouds can be treated as latent variables using the EM algorithm. Consequently, despite inaccuracies in their inference, the semantics improved the point cloud registration results. The proposed method also treats semantics as the observable variable; however, it can cope with the uncertainty of the object recognition because Dirichlet distribution is used to model the likelihood distribution.

We previously proposed the simultaneous localization and measurement-class estimation method in [25], [26]. In the method, mapped and unmapped object classes are simultaneously estimated using the environment map via the class-conditional measurement model. This model makes it possible to handle multiple object classes. However, the method cannot achieve probabilistic integration of the object recognition results. More specifically, the method cannot handle object recognition results as prior to estimate the measurement classes. In this work, we extended the method to handle multiple object classes that were estimated by an SL-based object recognition method. We also previously proposed localization methods combined with the DNN in [27]–[29]. The method proposed in this work is also combined with the DNN; however, the DNNs used in the previous works are not used for object recognition.

III. Proposed method

A. Graphical model and its posterior distribution

Figure 1 illustrates the graphical model of the proposed method. Our objective is to estimate the posterior distribution over the current vehicle pose denoted as

$$p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t}, \mathbf{m}, \Theta, D), \tag{1}$$

where t and $1:t$ represent current and sequential time data. The details of the variables are described in the second paragraph of Section I.

Because $\mathbf{c}_t$ depends on $\mathbf{x}_t$, this equation can be re-written using Bayes's theorem as

$$\begin{aligned} &p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t}, \mathbf{m}, \Theta, D) \\ =&\eta p(\mathbf{c}_t|\mathbf{x}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t-1}, \mathbf{m}, \Theta, D) \\ &\cdot p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t-1}, \mathbf{m}, \Theta, D), \end{aligned} \tag{2}$$

where η is a normalization constant. The first term of the right side of equation (2) can also be re-written using D-separation as

$$p(\mathbf{c}_t|\mathbf{x}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t-1}, \mathbf{m}, \Theta, D) = p(\mathbf{c}_t|\mathbf{x}_t, \mathbf{z}_t, \mathbf{m}, \Theta). \tag{3}$$

This distribution models the predictions of the object classes that use the SL-based object recognition with the condition where the vehicle pose, sensor measurement, map, and hyperparameters are given. We refer to this model as the *class prediction model (CPM)*, and the details are given in Section III-B.

The second term of the right side of equation (2) can also be re-written using the law of total probability and D-separation. Finally, the posterior distribution over the current vehicle pose is denoted as

$$\begin{aligned} &p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t}, \mathbf{m}, \Theta, D) \\ =&\eta p(\mathbf{c}_t|\mathbf{x}_t, \mathbf{z}_t, \mathbf{m}, \Theta) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) \\ &\cdot p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, \mathbf{c}_{1:t-1}, \mathbf{m}, \Theta, D) \mathrm{d}\mathbf{x}_{t-1}, \end{aligned} \tag{4}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$ is the motion model [30].

B. Class prediction model

The sensor measurement is denoted as $\mathbf{z}_t = (\mathbf{z}_t^1, \mathbf{z}_t^2, ..., \mathbf{z}_t^K)$, where K is the number of measurements. The probability over the object classes estimated using the SL-based object recognition is denoted as $p(\mathbf{c}_t) = (p(\mathbf{c}_t^1), p(\mathbf{c}_t^2), ..., p(\mathbf{c}_t^K))$, where $p(\mathbf{c}_t^k)$ is the corresponding discrete distribution to $\mathbf{z}_t^k$.

We first assume that each class prediction result is independent and factorize the CPM as

$$p(\mathbf{c}_t|\mathbf{x}_t, \mathbf{z}_t, \mathbf{m}, \Theta) = \prod_{k=1}^{K} p(\mathbf{c}_t^k|\mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta). \tag{5}$$

Consequently, we can separately model the CPM for each measurement. To define the CPM, we consider two cases in which the object class is correctly and incorrectly predicted, namely, the positive and negative classification cases. The CPM is represented using the linear combination of the two distributions over the cases expressed as

$$\begin{aligned} &p(\mathbf{c}_t^k|\mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta) \\ =&\begin{pmatrix} c_{\text{posi}} \\ c_{\text{nega}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{posi}}(\mathbf{c}_t^k|\mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta) \\ p_{\text{nega}}(\mathbf{c}_t^k|\mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta) \end{pmatrix}, \end{aligned} \tag{6}$$

where c_{posi} and c_{nega} are arbitrary constants satisfying $c_{\text{posi}} + c_{\text{nega}} = 1$, and $p_{\text{posi}}(\cdot)$ and $p_{\text{nega}}(\cdot)$ are distributions for modeling the positive and negative classifications, respectively.

Because the probability over the object classes is discrete, we model these distributions using Dirichlet distribution, $\text{Dir}(\cdot)$. For example, $p_{\text{posi}}(\cdot)$ is denoted as

$$\begin{aligned} &p_{\text{posi}}(\mathbf{c}_t^k|\mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta) \\ =&\text{Dir}\left(\mathbf{c}_t^k|\mathbf{a}(\mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta)\right), \\ =&\frac{\Gamma\left(\sum_{i\in C} {}^i a\right)}{\prod_{i\in C} \Gamma\left({}^i a\right)} \prod_{i\in C} p\left({}^i c_t^k\right)^{\left({}^i a - 1\right)}, \end{aligned} \tag{7}$$

where C is a list of the object classes, $\Gamma(\cdot)$ is the gamma function and $\mathbf{a}(\cdot) \in \mathcal{R}^{|C|}$, $(\mathbf{a}(\cdot))_i = {}^i a > 0$, are the hyperparameters of the Dirichlet distribution. $p_{\text{nega}}(\cdot)$ is also modeled using Dirichlet distribution.

To determine the hyperparameters, we can use the vehicle pose, sensor measurement, map, and hyperparameter for the SL-based object recognition. Therefore, we can determine the hyperparameters of the CPM using the hyperparameters of the SL-based object recognition, which are already obtained, i.e., the training has already been done. c_{posi} and c_{nega} shown in equation (6) can also be determined by considering the performance. Details of these parameter determinations are described in Section IV-B.

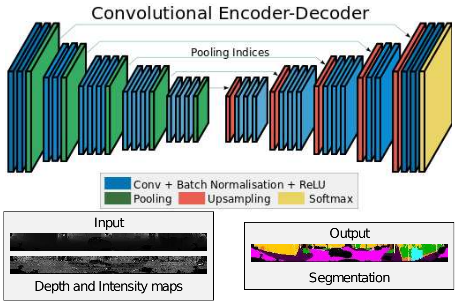

Fig. 2. The network of the point cloud SS. The architecture except for the input and output layers is the same as that of SegNet presented in [2].

IV. Implementation

In this work, we focus on the 3D LiDAR-based localization problem and present an implementation example of the proposed method using the PF and DNN-based point cloud SS. To validate the implemented method, we use the SemanticKITTI dataset [10]. This dataset provides the object labels of the 3D LiDAR measurements included in the KITTI odometry dataset [31]. This section describes the implementation details.

A. Object class estimation

1) Object classes: In the SemanticKITTI dataset, 34 classes, including 14 static objects, are provided. The 14 static objects are used because we aim to solve the localization problem. All the non-static objects are categorized as an unknown class. Totally, we use 15 classes denoted as $C \in \{\text{unknown}, C_{\text{static}}\}$, where C_{static} is a list of the static objects.

2) DNN-based SS: We implement the DNN-based point cloud SS as the SL-based object recognition. We referred to SegNet [2] to develop the network used in this study. Figure 2 illustrates the overview of the network.

We create image data using the 3D LiDAR measurement. Because each measurement, $\mathbf{z}_t^k$, contains a 3D point and intensity, we build depth and intensity maps; these maps are concatenated. The width and height of the maps, W and H, are set as $W = 360$ and $H = 32$, respectively. Hence, $W \times H \times 2$ size data is fed to the DNN.

The softmax layer is implemented as the activation function at the output layer. The DNN predicts discrete probabilistic distributions over the object classes of each pixel. The size of the output is $W \times H \times |C|$. The categorical cross entropy is used as the loss function

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{i\in C} p({}^i\hat{c}_n) \log p({}^i c_n), \tag{8}$$

where N is the number of data and $\hat{c} \in \{0, 1\}$ is the ground truth object class.

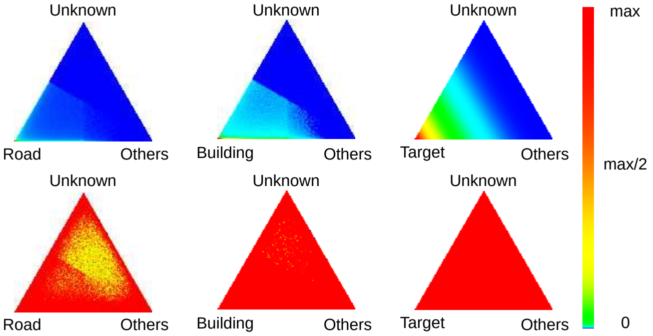

Fig. 3. Heatmaps of the SS results using the DNN (left and center) and the CPMs (right) with barycentric coordinate representation. The top and bottom figures are results of the positive and negative classifications.

B. Hyperparameter determination for CPM

As we mentioned in Section III-B, the vehicle pose, sensor measurement, map, and hyperparameters for the SL-based object recognition can be used to determine the hyperparameters for the CPM shown in equation (7). Because the hyperparameters for the SL-based object recognition are given, i.e., the DNN has already been trained, the DNN-based SS performance can be considered to determine the hyperparameters for the CPM. Therefore, we first conducted pre-experiments to investigate the performance.

1) Investigation of SS performance: Figure 3 shows the pre-experimental results. To obtain these results, we trained the DNN using sequences from 00 to 10 except for 08 of the SemanticKITTI dataset. The left and center figures are validation results using sequence 08. The figures illustrate heatmaps with barycentric coordinate representation which can visualize three-dimensional Dirichlet distribution. These 2D points, $\mathbf{p}$, are calculated as follows

$$\mathbf{p} = p_{\text{target}}\mathbf{p}_{\text{BL}} + p_{\text{others}}\mathbf{p}_{\text{BR}} + p_{\text{unknown}}\mathbf{p}_{\text{T}} \tag{9}$$

where p_{target}, p_{others}, and p_{unknown} are probabilities over the corresponding classes and $\mathbf{p}_{\text{BL}}$, $\mathbf{p}_{\text{BR}}$, and $\mathbf{p}_{\text{T}}$ are 2D points of the bottom left, right and top, respectively. p_{others} is the summation of the probabilities except for those of the target and unknown classes.

In the left and center cases of Fig. 3, road and building were set as target class. The top and bottom figures are results of the positive and negative classifications. In these classifications, the measurement is categorized as a class to that of the maximum probability. In the positive classification cases, the frequency of the target class side is higher than that of other sides. In the negative classification cases, the frequency of all of the areas is almost equivalent.

Based on these results, we assume that (1) the CPM is proportional to the object measurability in the positive classification case and (2) the CPM is uniform in the negative classification cases. Below, we discuss the calculation of the measurability and how the hyperparameters for the CPM are determined.

2) Measurability: The measurability of the i-th class object represents the possibility where the object is measured. Therefore, we define the measurability using the measurement models, denoted as $p(\mathbf{z}_t^k|\mathbf{x}_t, {}^i\mathbf{m})$, presented in [30]. Where ${}^i\mathbf{m}$ is the i-th class object map.

We define the measurability using the likelihood field model (LFM)

$$p_{\text{LFM}}(\mathbf{z}_t^k|\mathbf{x}_t, {}^i\mathbf{m}) = \begin{pmatrix} z_{\text{hit}} \\ z_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{hit}}(\mathbf{z}_t^k|\mathbf{x}_t, {}^i\mathbf{m}) \\ p_{\text{rand}}(\mathbf{z}_t^k|\mathbf{x}_t, {}^i\mathbf{m}) \end{pmatrix}, \tag{10}$$

where z_{hit} and z_{rand} are arbitrary constants satisfying $z_{\text{hit}} + z_{\text{rand}} = 1$, and $p_{\text{hit}}(\cdot)$ and $p_{\text{rand}}(\cdot)$ are used to model items related to the measurement of the mapped obstacles and random noise[1]. In the implementation, these parameters were set as $z_{\text{hit}} = 0.95$ and $z_{\text{rand}} = 0.05$.

In this work, we use the semantic map to calculate the LFM (or the semantic LFM described in later). Note that the CPM is calculated using the results of the LFM. The distance field (DF) representation that describes the nearest distance from obstacles to each voxel (or gird) is better solution to quickly calculate the LFM. Hence, in our implementation, i-th object map means a DF of i-th object and the semantic map means a set of the DFs. To efficiently implement the DFs, we used the method presented in [32].

$p_{\text{hit}}(\cdot)$ and $p_{\text{rand}}(\cdot)$, respectively, are denoted as

$$p_{\text{hit}}(\mathbf{z}_t^k|\mathbf{x}_t, {}^i\mathbf{m}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d(\mathbf{z}_t^k, \mathbf{x}_t, {}^i\mathbf{m})^2}{2\sigma^2}\right), \tag{11}$$

$$p_{\text{rand}}(\mathbf{z}_t^k|\mathbf{x}_t, {}^i\mathbf{m}) = \text{unif}(0, R), \tag{12}$$

where σ^2 is a measurement variance, R is the maximum measurement range, $d(\cdot)$ returns the closest distance from a scan point which is transformed based on the given pose to obstacles existing on the i-th object map, and $\text{unif}(\cdot)$ is the uniform distribution within a given range. σ^2 and R were set as $\sigma^2 = (0.1\ \text{m})^2$ and $R = 120$ m while considering specification of LiDAR used.

We define the unknown class object measurability using the exponential distribution

$$p_{\text{unknown}}(\mathbf{z}_t^k|\mathbf{x}_t, \mathbf{m}) = \frac{\lambda \exp\left(-\lambda r_t^k\right)}{1 - \exp\left(\lambda R\right)}, \tag{13}$$

where r_t^k is the measurement range and λ is its hyperparameter. λ is also determined while considering the LiDAR's specification, namely, ensuring the measurement possibility among the entire measurement range. Finally, it was set to 0.03.

3) Hyperparameter determination: As mentioned in Section IV-B.1, the i-th object class probability is proportional to the i-th object class measurability. The value of the Dirichlet distribution is high when ${}^ia$ and $p({}^ic_t^k)$ are higher than other values. Therefore, we also assume that ${}^ia$ is proportional to the measurability and determine the hyperparameter in the positive classification case as

$${}^ia(\mathbf{x}_t, \mathbf{z}_t^k, {}^i\mathbf{m}, \Theta) = 3m(\mathbf{x}_t, {}^ic_t^k, \mathbf{z}_t^k, \mathbf{m}) + 1. \tag{14}$$

[1] In [30]; the item with the maximum measurement value is considered. However, we do not consider the item because the 3D point cloud does not usually have such measurements.

$$m(\mathbf{x}_t, {}^{i}c_t^k, \mathbf{z}_t^k, \mathbf{m}) \\ = \begin{cases} p_{\text{unknown}}(\mathbf{z}_t^k | \mathbf{x}_t, \mathbf{m}) & (\text{if } {}^{i}c_t^k = \text{unknown}), \\ p_{\text{LFM}}(\mathbf{z}_t^k | \mathbf{x}_t, {}^{i}\mathbf{m}) & (\text{otherwise}), \end{cases} \tag{15}$$

where $m(\cdot)$ is the measurability. In the negative classification case, all the hyperparameters were set to one because we assume that the distribution is uniform, i.e., $p_{\text{nega}}(\mathbf{c}_t^k | \mathbf{x}_t, \mathbf{z}_t^k, \mathbf{m}, \Theta) = \text{Dir}(\mathbf{c}_t^k | \mathbf{1})$, where $\mathbf{1} \in \mathcal{R}^{|C|}$ is a vector that all elements are one. The right side of Fig. 3 shows the CPM of the positive (top) and negative (bottom) cases plotted using the above hyperparameters. Note that the values shown in equation (14) were experimentally determined while considering the actual classification results.

It is difficult to theoretically determine the parameters described above. The data-driven-based method using expectation-maximization algorithm to determine such parameters is presented in [30]. This method can be also applied to the proposed method. However, we empirically know that these parameters do not strongly affect the localization performance if these are roughly close to its optimal value. Therefore, we did not adopt the method in this work.

4) Coefficient determination: The coefficients of the CPM, c_{posi} and c_{nega}, shown in equation (6), are also determined based on the SS performance. Table I shows the SS accuracy (SSA). Based on these results, we set these coefficients as $c_{\text{posi}} = 0.7$ and $c_{\text{nega}} = 0.3$.

C. Particle filter-based posterior estimation

The target posterior distribution shown in equation (1) is estimated using the PF. The following procedures are recursively performed to estimate the joint posterior:

1) estimate the probabilities over the object classes
2) update the particles' poses based on the motion model
3) calculate the particles' likelihood using the CPM
4) estimate the vehicle pose and re-sample the particles

The details of 2) and 4) are given in [30]. In this study, the number of particles, M, was set to 500.

V. Simulation Experiments

We first compared the likelihood distributions calculated using three models in the simulation environment.

A. Conditions

1) Environment: A simple simulation environment was created as shown in Fig. 4, with two static and three dynamic object classes. The black area of the environment represents a free space. A color scheme is the same to that of the SemanticKITTI dataset.

2) Sensor measurement: A 2D LiDAR measurement with a maximum scan range of 80 m, scan angle of 190 deg, and scan angle resolution of 0.125 deg was simulated. White noise is added to the measurement ranges and angles.

3) Laser-wise object recognition: Object recognition is randomly performed for each measurement and its accuracy is controlled. If the object recognition is successful, the maximum probability of the corresponding class is set to approximately 0.9 and the other probabilities are set to approximately 0.05. If the object recognition fails, all the probabilities are randomly set. Because the summation of the randomly created discrete values is not to be one, these values are normalized to keep the condition of the probabilistic distribution.

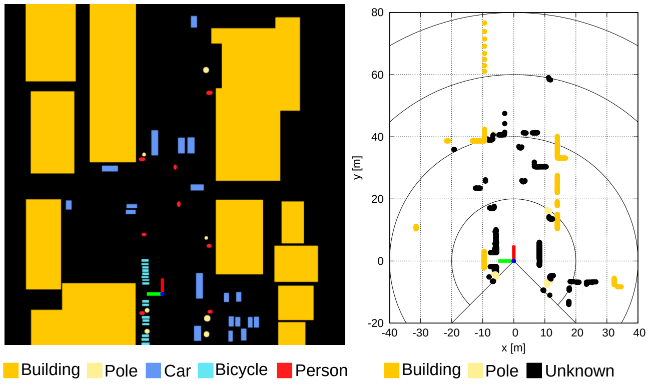

Fig. 4. Simulation environment (left) and ground truth scan and object recognition results (right). The non-static objects are categorized as an unknown class.

B. Comparison methods

1) Likelihood field model (LFM): Equation (10) is used to calculate the likelihood. Note that the object classes are not used, which indicates that all the points of the map are used for calculating the likelihood.

2) Semantic likelihood field model (SLFM): The object recognition results are simply used in the method. If a measurement is categorized as the i-th class object, equation (10) is used. If a measurement is categorized as the unknown class, equation (13) is used.

C. Simulation results

Figure 5 shows the simulation results. In the top, middle, and bottom cases, the object recognition accuracy was set to approximately 80 %, 50 %, and 20 %, respectively. The figures from left-to-right are the object recognition results and the likelihood distributions calculated using the LFM, SLFM, and CPM, respectively. The likelihood distributions were calculated around the ground truth, $(dx, dy) = (0, 0)$.

Because the LFM does not use the object recognition results, the likelihood calculation was not affected by the results. The SLFM was drastically affected by the object recognition results. Because wrong object labels were assigned to the measurement in the likelihood calculation using the SLFM, the object maps used for calculating the measurement model were not correctly selected. However, the CPM made it possible to robustly calculate the likelihood distributions to the segmentation accuracy degradation. In particular, the CPM generated the likelihood distributions more accurately than the LFM. These results showed that using the CPM achieves robust and accurate likelihood calculations even when environment and object recognition results are noisy. The accompanying video shows the comparison using the simulation.

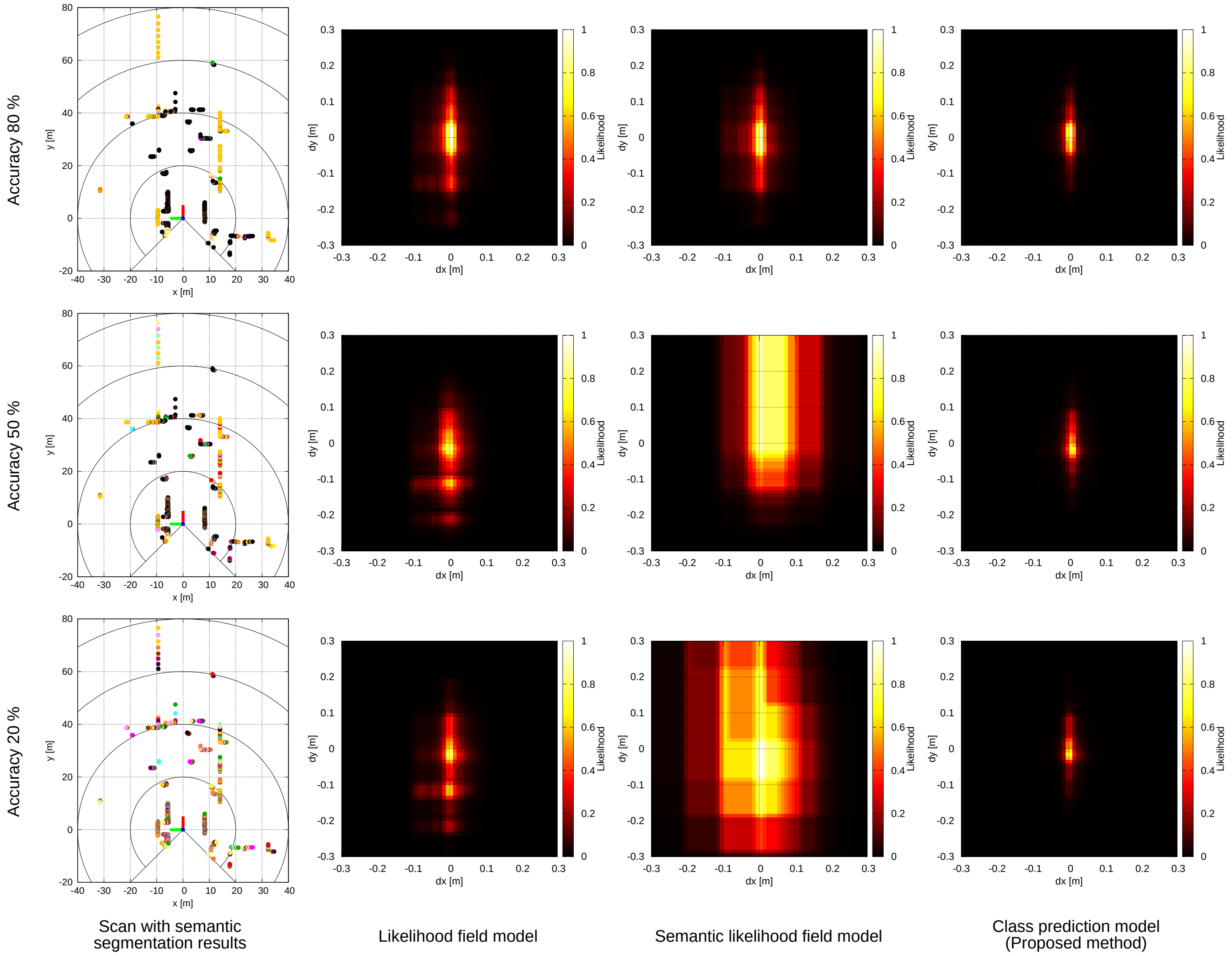

Fig. 5. Simulated scans, their object recognition results (left) and likelihood distributions around the ground truth, $(dx, dy) = (0, 0)$, calculated using the LFM, SLFM, and CPM, respectively. Object recognition accuracy was controlled and was set as approximately 80 %, 50 %, and 20 % in the top, middle, and bottom cases, respectively.

VI. Experiments with SemanticKITTI dataset

A. Map building

The SemanticKITTI dataset includes the ground truth trajectory of the vehicle. We first plotted the static object points according to the ground truth trajectory and built the semantic map. Note that we assumed that the ground surface is flat.

B. Noise control input simulation

The SemanticKITTI dataset does not include motion data, i.e., control input denoted as $\mathbf{u}_t = (\Delta d_t, \Delta\theta_t)$. Hence, we simulate odometry noise using the ground truth trajectory. We first compute the differences of the distance and heading angle, $\Delta\hat{d}_t$ and $\Delta\hat{\theta}_t$, between consecutive frames and then add noise as

$$\Delta d_t \sim \mathcal{N}(\gamma_{\text{dist}} \Delta\hat{d}_t, \sigma^2_{\text{dist}}), \tag{16}$$

$$\Delta\theta_t \sim \mathcal{N}(\gamma_{\text{angle}} \Delta\hat{\theta}_t, \sigma^2_{\text{angle}}), \tag{17}$$

where $\mathcal{N}(a, b^2)$ is Gaussian with mean a and variance b^2, and γ and σ are arbitrary constants. These arbitrary constants were set as $\gamma_{\text{dist}} = 0.99$, $\gamma_{\text{angle}} = 1.01$, $\sigma^2_{\text{dist}} = (0.01 \text{ m})^2$, and $\sigma^2_{\text{angle}} = (0.01 \text{ deg})^2$. To simulate symmetric errors, γ_{dist} and γ_{angle} are used.

C. Point cloud semantic segmentation

The SemanticKITTI dataset opens 11 sequences; however, we used seven sequences for the validation because of memory limitation. The SS network is trained using sequences except for the target sequence, i.e., 10 sequences are used for the training.

D. Results

We compared the pose estimation accuracy of three PF-based localization methods which, respectively, use the CPM, LFM, and SLFM for the likelihood calculation. The CPM is the proposed method. The LFM does not use the SS results and is described in Section V-B.1. The SLFM simply uses the SS results for calculating the likelihoods and is described in Section V-B.2. To evaluate localization accuracy, we compared the localization results of every frame with the ground truth trajectory and measured the position and yaw angle estimation errors on the xy plane over the whole trajectory.

Table I lists the estimation errors, static points rate (SPR), and SS accuracy (SSA). Here, the SPR and SSA are the mean rate of the static objects and accuracy of the SS in one LiDAR measurement.

From the estimation errors, it was observed that the proposed method achieved the most accurate and precise pose estimation. Although the SS accuracy was not usually high, (in particular, the minimum accuracy was less than 50 % in sequence 10), the pose estimation using the proposed method worked robustly. However, the pose estimation accuracy using the SLFM was not accurate.

In sequences 06 and 07, the SPRs are lower than those of the other sequences, i.e., these sequences contain many moving objects like cars. Because the LFM does not consider any environment changes as described in equation (10), its estimation accuracy was slightly degraded in these sequences. Particularly in sequence 06, the SS accuracy was also low and the estimate using the SLFM did not work well. However, the proposed method could accurately perform the estimation in these sequences. From the experimental results, we confirmed that the proposed method could accurately and robustly perform localization even when the SS accuracy is low and the environment is dynamic.

E. Drawbacks

Although the proposed method works better than other methods, it has some drawbacks. The proposed method requires large memory and more computational costs than the others. Because several object maps are used in the proposed method, the memory cost requirement for allocating the maps is obviously larger than that of the LFM-based method.

The PF-based localization wastes more time in the likelihood calculation process. If the number of particles are set to M, the computational complexity of the process of LFM and SLFM is $O(M)$. However, the computational complexity of the proposed method is $O(M|C|)$, where $|C|$ is the number of the object classes.

Table II shows a comparison of the computational times of the process in sequence 03. The proposed method required considerably more time than the other methods. Additionally, improving the computational speed by enhancing the algorithms is difficult. To improve the speed, GPU implementation is necessary.

The proposed method also requires an object recognition method which estimates the probabilistic distribution over the object classes for each measurement. To address this requirement, we used the DNN. However, using the DNN also requires a large computational cost. Note that the results shown in Table II do not include the processing time of the DNN. Conditional random fields can be applied to such object recognition [33]; however, its computation is also inefficient because it requires an iterative process such as loopy belief propagation [34] to estimate the probability. Efficient probability estimation is also necessary for real-time application of the proposed method.

VII. Conclusion

This paper has presented a localization method for leveraging laser-wise probabilistic object recognition. Supervised learning, which provides the probabilities over the measured object classes, was integrated into the probabilistic localization framework. The proposed method used Dirichlet distribution to calculate the likelihood, making it possible to cope with the uncertainty of object recognition. From the experiments, we showed that (1) the proposed method accurately generated likelihood distribution even when the SS accuracy was inaccurate and (2) the estimation accuracy using the proposed method was highest in the proposed method than in LFM- and SLFM-based localization methods.

Because the computational cost of the proposed method is quite large, our future work will focus on improving the cost by applying GPU implementation and efficient object recognition. In addition, we plan to leverage the semantics for detection of localization failures as presented in [35].

ACKNOWLEDGMENT

This work was supported by JST COI under Grant JP-MJCE1317 and KAKENHI under Grant 40786092.

References

[1] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[3] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. pages 4558–4567, 2017.

[4] A. Dewan, G. L. Oliveira, and W. Burgard. Deep semantic classification for 3D LiDAR data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3544–3549, 2017.

[5] J. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018.

[6] E. Stenborg, C. Toft, and L. Hammarstrand. Long-term visual localization using semantically segmented images. pages 6484–6490, 2018.

[7] V. Vaquero, K. Fischer, F. Moreno-Noguer, A. Sanfeliu, and S. Milz. Improving map re-localization with deep 'movable' objects segmentation on 3D LiDAR point clouds. *arXiv:1910.03336*, 2019.

[8] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss. SuMa++: Efficient LiDAR-based semantic SLAM. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4530–4537, 2019.

[9] A. Zaganidis, A. Zerntev, T. Duckett, and G. Cielniak. Semantically assisted loop closure in SLAM using NDT histograms. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4562–4568, 2019.

[10] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[11] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

TABLE I

Position / angle estimation errors using the CPM, LFM, and SLFM, static points rate (SPR), and semantic segmentation accuracy (SSA). Units of the position and angle estimation errors are centimeters and degrees, respectively.

	Sequence	03	04	05	06	07	09	10
CPM	Ave	7.34 / 0.20	6.64 / 0.14	5.42 / 0.15	8.98 / 0.20	6.56 / 0.18	7.55 / 0.18	5.52 / 0.17
	Std	5.28 / 0.18	5.28 / 0.12	3.38 / 0.15	6.91 / 0.19	5.01 / 0.20	5.13 / 0.17	3.70 / 0.15
	Max	42.34 / 1.12	36.90 / 0.74	28.36 / 1.62	67.07 / 1.36	39.71 / 1.71	69.89 / 1.17	32.91 / 1.19
LFM	Ave	11.12 / 0.26	10.60 / 0.29	7.40 / 0.18	13.88 / 0.32	14.55 / 0.26	9.63 / 0.22	5.93 / 0.18
	Std	7.31 / 0.25	9.98 / 0.24	5.30 / 0.17	14.66 / 0.35	7.52 / 0.24	7.88 / 0.22	3.96 / 0.17
	Max	44.50 / 1.44	73.11 / 1.27	44.20 / 1.52	133.60 / 3.54	56.09 / 1.83	120.20 / 3.89	45.78 / 1.25
SLFM	Ave	11.00 / 0.33	14.74 / 0.26	9.29 / 0.26	17.88 / 0.42	11.23 / 0.33	14.06 / 0.39	9.78 / 0.33
	Std	7.33 / 0.28	12.24 / 0.24	6.80 / 0.25	13.56 / 0.40	7.52 / 0.29	10.86 / 0.35	7.02 / 0.30
	Max	43.90 / 1.52	77.01 / 1.88	65.46 / 2.44	100.39 / 2.69	64.94 / 2.20	104.53 / 3.32	86.76 / 2.29
SPR	Ave	95.92 %	95.74 %	96.64 %	89.26 %	87.08 %	93.79 %	95.92 %
	Std	2.46 %	2.96 %	2.30 %	6.85 %	7.81 %	4.74 %	3.88 %
	Min	84.75 %	80.62 %	89.79 %	59.37 %	57.97 %	65.66 %	70.22 %
	Max	99.66 %	99.11 %	98.81 %	97.48 %	98.33 %	99.29 %	99.67 %
SSA	Ave	76.90 %	77.35 %	79.04 %	74.32 %	85.30 %	78.61 %	75.21 %
	Std	6.50 %	5.65 %	4.07 %	6.76 %	4.67 %	5.30 %	5.80 %
	Min	60.33 %	60.17 %	63.49 %	54.71 %	70.51 %	57.36 %	44.80 %
	Max	92.10 %	91.02 %	89.84 %	86.16 %	95.07 %	90.41 %	85.99 %

TABLE II

Computation times in milliseconds.

	Ave	Std	Min	Max
CPM	101.9	4.6	83	117
LFM	16.1	1.0	9	19
SLFM	18.8	1.4	13	23

[12] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings of the IEEE/RSJ Intelligent Robots and Systems*, pages 2743–2748, 2003.

[13] N. Akai, K. Inoue, and K. Ozaki. Autonomous navigation based on magnetic and geometric landmarks on environmental structure in real world. *Journal of Robotics and Mechatronics*, 26(2):158–165, 2014.

[14] N. Akai, L. Y. Morales, T. Yamaguchi, E. Takeuchi, Y. Yoshihara, H. Okuda, T. Suzuki, and Y. Ninomiya. Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 1147–1152, 2017.

[15] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[16] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.

[17] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya. Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1357–1364, 2017.

[18] A. Zaganidis, M. Magnusson, T. Duckett, and G. Cielniak. Semantic-assisted 3D normal distributions transform for scan registration in environments with limited structure. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4064–4069, 2017.

[19] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak. Integrating deep semantic segmentation into 3-D point cloud registration. *IEEE Robotics and Automation Letters*, 3(4):2942–2949, 2018.

[20] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3D mapping with semantic knowledge. *Robot Soccer World Cup*, pages 335–346, 2005.

[21] C. Yi, I. H. Suh, G. H. Lim, and B. Choi. Active-semantic localization with a single consumer-grade camera. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 2161–2166, 2009.

[22] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas. Localization from semantic observations via the matrix permanent. *The International Journal of Robotics Research*, 35(1–3):73–99, 2016.

[23] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1722–1729, 2017.

[24] S. A. Parkison, L. Gan, M. G. Jadidi, and R. M. Eustice. Semantic iterative closest point through expectation-maximization. In *Proceedings of the British Machine Vision Conference*, 2018.

[25] N. Akai, L. Y. Morales, and H. Murase. Mobile robot localization considering class of sensor observations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3159–3166, 2018.

[26] N. Akai, L. Y. Morales, T. Hirayama, and H. Murase. Toward localization-based automated driving in highly dynamic environments: Comparison and discussion of observation models. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 2215–2222, 2018.

[27] N. Akai, L. Y. Morales, and H. Murase. Simultaneous pose and reliability estimation using convolutional neural network and Rao-Blackwellized particle filter. *Advanced Robotics*, 32(17):930–944, 2018.

[28] N. Akai, L. Y. Morales, and H. Murase. Reliability estimation of vehicle localization result. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 740–747, 2018.

[29] N. Akai, T. Hirayama, and H. Murase. Hybrid localization using model- and learning-based methods: Fusion of Monte Carlo and E2E localizations via importance sampling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6469–6475, 2020.

[30] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[31] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[32] N. Akai, T. Hirayama, and H. Murase. 3D Monte Carlo localization with efficient distance field representation for automated driving in dynamic environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2020 (accepted).

[33] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*, pages 1097–1104. 2005.

[34] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[35] N. Akai, L. Y. Morales, T. Hirayama, and H. Murase. Misalignment recognition using Markov random fields with fully connected latent variables for detecting localization failures. *IEEE Robotics and Automation Letters*, 4(4):3955–3962, 2019.