2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Hybrid Localization using Model- and Learning-Based Methods: Fusion of Monte Carlo and E2E Localizations via Importance Sampling

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

***Abstract*— This paper proposes a hybrid localization method that fuses Monte Carlo localization (MCL) and convolutional neural network (CNN)-based end-to-end (E2E) localization. MCL is based on particle filter and requires proposal distributions to sample the particles. The proposal distribution is generally predicted using a motion model. However, because the motion model cannot handle unanticipated errors, the predicted distribution is sometimes inaccurate. The use of other ideal proposal distributions, such as the measurement model, can improve robustness against such unanticipated errors. This technique is called importance sampling (IS). However, it is difficult to sample the particles from such ideal distributions because they are not represented in the closed form. Recent works have proved that CNNs with dropout layers represent the posterior distributions over their outputs conditioned on the inputs and the CNN predictions are equivalent to sampling the outputs from the posterior. Therefore, the proposed method utilizes a CNN to sample the particles and fuses them with MCL via IS. Consequently, the advantages of both MCL and E2E localization can be simultaneously leveraged while preventing their disadvantages. Experiments demonstrate that the proposed method can smoothly estimate the robot pose, similar to the model-based method, and quickly re-localize it from the failures, similar to the learning-based method.**

I. Introduction

In autonomous navigation, accurate and robust localization is of utmost importance, e.g., [1]–[3]. Many authors have tackled the localization problem, and various model- and learning-based localization methods have been proposed thus far. However, the results of current state-of-the-art localization methods cannot be considered ideal owing to multiple factors. Model-based methods are weak toward unanticipated errors such as wheel slippage because these errors cannot be exactly modeled. On the other hand, learning-based methods can cope with unanticipated errors if a training dataset encapsulates such errors. However, the inferences derived using learning-based methods are usually unstable. Additionally, it is difficult to achieve precise autonomous navigation with only learning-based methods. Localization performance can be improved if the model- and learning-based methods are effectively fused, but a naive approach leads to degradation in the localization performance. This paper proposes a novel approach to fuse model- and learning-based localization methods that leverages the advantages from both methods while mitigating their disadvantages.

[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

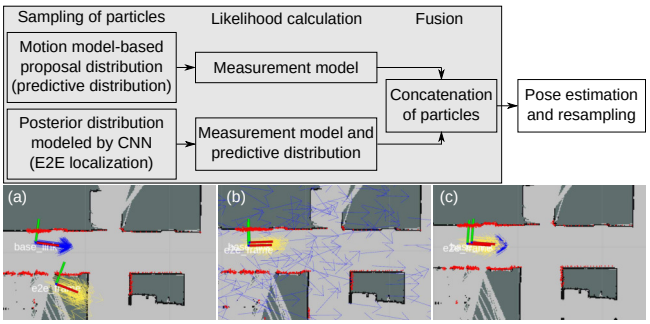

Fig. 1. Top and bottom figures show the overview of the proposed method and examples of successful estimates by the method. Blue and yellow arrows illustrate the model- and learning-based particles. In (a), localization is successful even though the E2E localization has failed. (b) and (c) illustrate the fast re-localization performance, where (b) shows the initial state, i.e., the initial pose is not given, and (c) shows the estimation result after slightly moving from the pose in (b).

We propose a hybrid localization method that fuses Monte Carlo localization (MCL) [4], [5] and convolutional neural network (CNN [6])-based end-to-end (E2E) localization. MCL is based on particle filter (PF) and requires proposal distributions to sample the particles. The proposal distribution is generally predicted using a motion model, and the predicted distribution is referred to as *predictive distribution* [7]. Because unanticipated errors cannot be exactly modeled, the predictive distribution is occasionally inaccurate. The use of other ideal proposal distributions, such as the measurement model, can improve robustness against such unanticipated errors. This technique is referred to as importance sampling (IS) [8]. However, it is difficult to sample the particles from such ideal distributions because they are not represented in the closed form.

Recent works have proved that CNNs with dropout layers represent the posterior distributions over their outputs conditioned on the inputs and the CNN predictions are equivalent to sampling the outputs from the posterior [9]–[11]. Therefore, the proposed method utilizes a CNN to sample the particles and fuses them with MCL via IS.

The top and gray parts of Fig. 1 illustrate an overview of the proposed method and the fusion process via IS, respectively. In the proposed method, two types of particles, referred to as *model- and learning-based particles*, are used. They are derived from the predictive distribution and posterior distribution which are modeled using the CNN, respectively. The bottom figures illustrate examples of successful estimates obtained using the proposed method in the 2D LiDAR localization problem; the LiDAR measurements are depicted in red. It should be noted that learning-based

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

particle (yellow arrows) sampling significantly affected the changes in the direction of the robot.

In (a), the localization was successful despite the mis-convergence of the learning-based particles because the model-based particles (blue arrows) correctly converged. (b) and (c) show the fast re-localization performance. (b) shows the initial state, i.e., the model-based particles were randomly sampled because no previous information regarding their sampling was available. However, the learning-based particles correctly converged. (c) shows the result of robot-pose estimation after slightly moving from the pose in (b). Both the model- and learning-based particles correctly converged in (c) because the learning-based particles are fused. Experiments demonstrated that the proposed method can smoothly estimate the robot pose, similar to the model-based method, and quickly re-localize it from the failures, similar to the learning-based method. The contribution of this study is the proposal of a method that fuses MCL and E2E localization via IS.

The rest of this paper is organized as follows. Section II summarizes the related work. Sections III and IV detail the proposed method and its implementation in the 2D localization problem, respectively. Section V describes the experimental conditions and results. Section VI concludes this work.

II. Related work

The MCL [4], [5] has been applied to various types of localization problems, and several extensions have been proposed, such as adaptive MCL [12] and augmented MCL [13]. Thrun *et al*. [14] proposed mixture MCL that uses multiple proposal distributions. In [14], three approaches were presented. The importance factor, i.e., the likelihood, was calculated using the motion-model, predictive-distribution, and distribution over the previous pose, respectively. Note that the first approach was not implemented in [14] because it cannot cope with the global-localization problem.

All the three approaches presented in [14] use the measurement model for particle sampling; however, this sampling is complex and inaccurate. The authors mentioned that the key method for the sampling involved learning a joint distribution sampling model over the pose and the measurement from data. In [14], the authors collected both the pose and measurement data using a real robot and created trees that permitted fast sampling with compressed measurement features. Although this method enables efficient particle sampling, hwever it has two problems; 1) the tree significantly increases the memory requirements and 2) drawing the particles on a correct area is not reliable. Particularly, mapping the sensor measurements to a low-dimensional feature vector significantly decreases the sampling accuracy.

Lenser and Veloso [15] proposed a failure recovering method for the MCL called sensor resetting (SR). SR also uses a sampling method based on the measurement model to recover from the failures if mis-convergence is detected. SR detects the failures based on the threshold. However, SR easily occurs mis-convergence again if the particles are drawn into incorrect areas because SR does not use IS, i.e., all particles are evaluated using only the measurement model. Hence, the threshold for detecting the failures must be carefully determined. Ueda *et al*. [16] proposed expansion resetting (ER), which is similar to SR and suitable for recovering the position tracking failures. However, ER also requires the exact threshold to detect the failures because it does not use IS either.

Recent works regarding CNNs have significantly improved the data-driven estimation methods. Kendall *et al*. [17] proposed PoseNet, which is an E2E localization method and learns the relationship between the camera image and pose. Charles *et al*. [18], [19] proposed PointNet and PointNet++, which enables E2E point cloud handling. Inspired by these works, various types of CNN-based-localization and point-registration methods have been proposed [20]–[26]. Additionally, Lu *et al*. [27]. proposed a localization method that replaces the processes involved in the localization framework to three networks. Amini *et al*. [28] presented a method that uses the outputs of the E2E navigation network, i.e., probabilistic distributions over a steering angle, for localizing an ego-vehicle pose. These studies have demonstrated the significant localization performance of the CNNs.

Kendall *et al*. [10] also asserted that the uncertainty in pose estimation using a CNN can be evaluated if the CNN has dropout layers. This uncertainty-estimation method is referred to as Monte Carlo (MC) dropout. MC dropout can be applied to various types of applications [29], [30]. With respect to MC dropout, several uncertainty-estimation methods have been recently formulated [31], [32]. These uncertainty-estimation methods can increase the feasibility of CNN-based applications. However, Sattler *et al*. [33] reported the limitations of CNN-based pose regression, i.e., there is no guarantee in the CNN-based pose regression, unlike structure-based approaches, generalize beyond their training data. Therefore, we believe that an effective strategy to fuse the E2E localization method with the model-based localization method is necessary to utilize E2E localization for autonomous navigation. To the best of our knowledge, this is the first time that E2E localization has been fused with MCL via IS.

We previously proposed a localization and its reliability-estimation method that used MCL and a CNN in [34], [35]. The CNN was used for detecting the localization failures and its output was used to calculate the likelihood of the particles. However, this method did not fuse the pose-estimation results by MCL and CNN and could not utilize the advantages to improve robustness. Learning-based failure detection methods for the localization and point cloud registration have also been recently proposed [36], [37]; however, these methods also do not utilize the inferences to improve robustness. The method proposed in this study improves robustness by fusing both the model- and learning-based methods.

III. Proposed method

This section describes MCL and the fusion method for the E2E localization with MCL via IS.

A. MCL

MCL is based on PF and estimates the posterior distribution over the current pose, $\mathbf{x}_t$, with conditions that are sequences of the control inputs and sensor measurements, $\mathbf{u}_{1:t}$ and $\mathbf{z}_{1:t}$, and a map $\mathbf{m}$. The posterior distribution can be recursively estimated as

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

where η is a normalization constant, $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$ is the measurement model, and $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$ is the motion model. For notational convenience, we define $\overline{Bel}(\mathbf{x}_{t-1})$ as follows.

$$\begin{aligned} &\overline{Bel}(\mathbf{x}_{t-1}) \\ &\stackrel{\text{def}}{=} \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, \mathbf{m}) d\mathbf{x}_{t-1}. \end{aligned} \tag{2}$$

$\overline{Bel}(\mathbf{x}_{t-1})$ is the predictive distribution [7] and is used as the proposal distribution. The particles derived from the distribution, ${}^m\mathbf{s}_t$, are referred to as *model-based particles* in this work.

In PF, the likelihood of the particles is denoted by the quotient of the target and the proposal distribution. The target distribution for MCL is denoted by $p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{m})$. Hence, the likelihood for the model-based particles, ${}^m l_t$, is calculated as follows.

$${}^m l_t = \eta \frac{p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) \overline{Bel}(\mathbf{x}_{t-1})}{\overline{Bel}(\mathbf{x}_{t-1})} = \eta p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}). \tag{3}$$

B. Fusion of E2E localization with MCL

Based on the recent works presented in [9]–[11], the inference using the CNN-based E2E localization with the dropout layers can be represented as follows.

$${}^l\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{z}_t). \tag{4}$$

Namely, the inferences are equivalent to sampling the pose from the posterior distribution. Therefore, we used the posterior as the proposal distribution and referred to the particles generated from the distribution, ${}^l\mathbf{s}_t$, as *learning-based particles*.

The posterior can be approximately represented using a finite number of particles as

$$p(\mathbf{x}_t|\mathbf{z}_t) \simeq \frac{1}{{}^l M} \sum_{i=1}^{{}^l M} \delta(\mathbf{x} - {}^l\mathbf{x}_t^{[i]}), \tag{5}$$

where δ is the Kronecker delta, which is 1 when the values within the bracket are equal and 0 otherwise, and ${}^l M$ is the number of learning-based particles. The likelihood of the learning-based particles can be calculated as follows.

$$\begin{aligned} {}^l l_t =& \eta \frac{p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})}{p(\mathbf{x}_t|\mathbf{z}_t)} \overline{Bel}(\mathbf{x}_{t-1}), \\ \simeq& \eta {}^l M \frac{p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})}{\sum_{i=1}^{{}^l M} \delta(\mathbf{x} - {}^l\mathbf{x}_t^{[i]})} \overline{Bel}(\mathbf{x}_{t-1}). \end{aligned} \tag{6}$$

To fuse E2E localization with MCL, the model- and learning-based particles are concatenated after calculating the likelihood. The concatenated particles, $\mathbf{s}_t = ({}^m\mathbf{s}_t, {}^l\mathbf{s}_t)$, approximately represent the posterior over the current pose described in equation (1). Finally, the current pose is estimated as the expectation of the posterior, i.e., the weighted average of both the particle sets.

IV. Implementation

This section describes the implementation of the proposed method in the 2D LiDAR-based localization problem.

A. Variable definitions

The objective is to estimate the current pose, $\mathbf{x}_t$, which is composed of the 2D position, x_t and y_t, and heading direction, θ_t, on the given map, $\mathbf{m}$, modeled by a 2D occupancy grid map. The 2D LiDAR measurement is represented by $\mathbf{z}_t = (\mathbf{z}_t^1, \mathbf{z}_t^2, ..., \mathbf{z}_t^K)$, where K is the number of measurements and $\mathbf{z}_t^k$ is composed of the measurement range, r_t^k, and angle, ψ_t^k. Hokuyo LiDAR (UTM-30LX) is used in this work and its specifications are as follows. 1) The measurement angle is 270 degrees, 2) the measurement angle resolution is 0.25 degrees, and 3) the maximum measurement range is 30 m. The robots used in this work are equipped with wheel encoders. $\mathbf{u}_t = (\Delta d_t, \Delta\theta_t)$ is the control input, and Δd_t and $\Delta\theta_t$, respectively, are the translation distance and heading direction from $t-1$ to t measured by the encoders.

B. Model-based MCL

1) Motion model: Let $\mathbf{s}_{t-1}$ and ${}^m\mathbf{s}_{t-1}$ be the particles that approximately represent the posterior over the pose at $t-1$ and are re-sampled from the posterior. Here, ${}^m\mathbf{s}_{t-1}$, composed of ${}^m M$ particles, is used as the previous states for the model-based particles. In this study, ${}^m M$ was set to 500. The pose of the model-based particles is first updated on the basis of the motion model as

$$\begin{pmatrix} {}^m x_t^{[i]} \\ {}^m y_t^{[i]} \\ {}^m \theta_t^{[i]} \end{pmatrix} = \begin{pmatrix} {}^m x_{t-1}^{[i]} \\ {}^m y_{t-1}^{[i]} \\ {}^m \theta_{t-1}^{[i]} \end{pmatrix} + \begin{pmatrix} \Delta d_t^{[i]} \cos {}^m\theta_{t-1}^{[i]} \\ \Delta d_t^{[i]} \sin {}^m\theta_{t-1}^{[i]} \\ \Delta\theta_t^{[i]} \end{pmatrix}, \tag{7}$$

$$\Delta d_t^{[i]} \sim \mathcal{N}(\Delta d_t, \alpha_1 \Delta d_t^2 + \alpha_2 \Delta\theta_t^2), \tag{8}$$

$$\Delta\theta_t^{[i]} \sim \mathcal{N}(\Delta\theta_t, \alpha_3 \Delta d_t^2 + \alpha_4 \Delta\theta_t^2), \tag{9}$$

where α_{1-4} are the arbitrary constants to represent the uncertainty of encoder measurements and $\mathcal{N}(a, b^2)$ is Gaussian with mean, a, and variance, b^2. The arbitrary parameters were set as follows: $\alpha_1 = 0.025$, $\alpha_2 = 0.0125$, $\alpha_3 = 0.0325$, and $\alpha_4 = 0.0615$.

2) Measurement model: After updating the particles using the motion model, their likelihoods are calculated using the measurement model. Because we assume that the measurements are independent of one another, the likelihood of the model-based particles is calculated as

$${}^m\omega_t^{[i]} = \prod_{k=1}^{K} p(\mathbf{z}_t^k|{}^m\mathbf{x}_t^{[i]}, \mathbf{m}). \tag{10}$$

Each measurement is modeled using the likelihood field model [5].

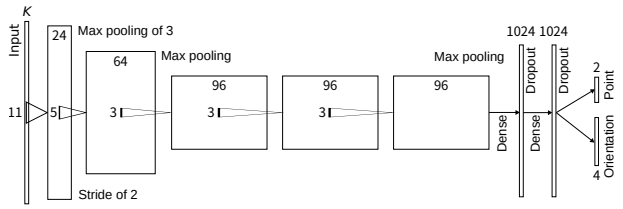

Fig. 2. Architecture of the CNN.

C. E2E localization and fusion via IS

1) CNN-based E2E localization: The architecture of the CNN used in this study is depicted in Fig. 2. We referred to AlexNet [38] with slight modifications to build the architecture. The measurement-range vector $\mathbf{r}_t = (r_t^1, r_t^2, ..., r_t^K)$ is fed to the CNN and the current 2D position, $\mathbf{p}_t$, and quaternion, $\mathbf{q}_t$, are predicted. The dropout layers are implemented after implementing the fully connected (dense) layers; the dropout rate was set to 10 %. Notably, we denote the posterior modeled using the CNN by $p(\mathbf{x}_t|\mathbf{z}_t)$ even the measurement-range vector is only fed to the CNN.

The loss function of the CNN is defined with respect to the literature [17] as

$$\mathcal{L}(\mathbf{z}_t) = \kappa\|\hat{\mathbf{p}}_t - \mathbf{p}_t\|_2 + (1-\kappa)\left\|\hat{\mathbf{q}}_t - \frac{\mathbf{q}_t}{\|\mathbf{q}_t\|_2}\right\|_2, \quad (11)$$

where $\hat{\mathbf{p}}_t$ and $\hat{\mathbf{q}}_t$ are the ground truth position and quaternion, respectively, and κ is an arbitrary constant that satisfies $0 \leq \kappa \leq 1$. In implementation, κ was set to 0.01. Because the norm of the quaternion must be one, the predicted quaternion was normalized.

In the localization phase, the same sensor measurement is fed to the CNN with different dropout neurons and ${}^lM$ poses are drawn. ${}^lM$ was set to 100. We used GeForce GTX TITAN X as the GPU and implemented the CNN using Keras [39]. The average prediction time taken by the CNN for the 100 particles was approximately 25 msec.

2) Likelihood calculation: As shown in equation (6), the likelihood of the learning-based particles is calculated using the measurement model, $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$, and the predictive distribution, $\overline{Bel}(\mathbf{x}_{t-1})$. The predictive distribution is approximately represented by the model-based particles updated using the motion model presented in equation (7). To define the probabilistic distribution based on the particles, we use a kernel-density estimation. The probability of the predictive distribution is approximated as

$$\begin{aligned}&\overline{Bel}({}^l\mathbf{x}_t^{[i]}; \mathbf{x}_{t-1}) \\ &\simeq \tau\frac{1}{{}^mM}\sum_{i=1}^{{}^mM}\mathcal{N}({}^l\mathbf{x}_t^{[i]}; {}^m\mathbf{x}_t^{[i]}, \Sigma) + (1-\tau)\mathrm{unif}(\mathbf{x}),\end{aligned} \quad (12)$$

where τ is an arbitrary constant satisfying $0 \leq \tau \leq 1$, Σ is a covariance matrix, and $\mathrm{unif}(\mathbf{x})$ is a uniform distribution defined within the areas that the robot can exist. If τ is set to one, the global-localization problem cannot be resolved because the likelihood of the particles far from the model-based particles will be vanished. In implementation, τ was set to 0.95 and Σ was set to $\mathrm{diag} = (10^{-4}\ \mathrm{m}^2, 10^{-4}\ \mathrm{m}^2, 10^{-2}\ \mathrm{deg}^2)$.

The likelihood of the learning-based particles is calculated as follows.

$${}^l\omega_t^{[i]} = {}^lM\prod_{k=1}^{K} p(\mathbf{z}_t^k|{}^l\mathbf{x}_t^{[i]}, \mathbf{m})\overline{Bel}({}^l\mathbf{x}_t^{[i]}; \mathbf{x}_{t-1}). \quad (13)$$

The inference derived using the CNN is usually unstable. However, the proposed method can prevent the effect of unstable estimation because the likelihood of the learning-based particles is calculated using the predictive distribution.

D. Pose estimation and re-sampling

Before estimating the current pose, the model- and learning-based particles are concatenated, $\mathbf{s}_t = ({}^m\mathbf{s}_t, {}^l\mathbf{s}_t)$, and the weight of all particles is normalized as follows.

$${}^m\omega_t^{[i]} \leftarrow \frac{{}^m\omega_t^{[i]}}{\sum_{j=1}^{{}^mM} {}^m\omega_t^{[j]} + \sum_{j=1}^{{}^lM} {}^l\omega_t^{[j]}}, \quad (14)$$

$${}^l\omega_t^{[i]} \leftarrow \frac{{}^l\omega_t^{[i]}}{\sum_{j=1}^{{}^mM} {}^m\omega_t^{[j]} + \sum_{j=1}^{{}^lM} {}^l\omega_t^{[j]}}. \quad (15)$$

Then, the current pose is estimated as follows.

$$\mathbf{x}_t = \sum_{i=1}^{{}^mM} {}^m\omega_t^{[i]}{}^m\mathbf{x}_t^{[i]} + \sum_{i=1}^{{}^lM} {}^l\omega_t^{[i]}{}^l\mathbf{x}_t^{[i]}. \quad (16)$$

Finally, ${}^mM$ particles are re-sampled based on the concatenated particles. The estimated pose at this instance is used for the evaluation described in Section V.

V. Experiments

A. Experimental conditions

1) Simulation environment and training dataset: We first operated a real robot and built consistent maps using 2D LiDAR SLAM [40]. The consistent maps were used to simulate the LiDAR measurements. Because we used the Hokuyo LiDAR (UTM-30LX) in the real experiments, we simulated its measurements. The white noise according to Gaussian with a variance of $0.025\ \mathrm{m}^2$ was added to each measurement range. Because the simulation allows us to obtain the ground truth pose, $\hat{\mathbf{x}}$, a training dataset, $D = \{\hat{\mathbf{x}}^{(n)}, \mathbf{z}^{(n)}|n = 1, ..., N\}$, can be easily created. In both the simulation and real experiments, the CNN was only trained on the datasets created using the simulation.

2) CNN: We first created a dataset in a large car-garage environment to validate E2E localization. A total of 2500 training samples and 500 test samples were collected. We also created a dataset that included the same number of samples in a dynamic simulation environment. In the dynamic environment, 40 randomly walking obstacles were simulated.

Table I lists the average, standard deviation, minimum, and maximum estimation errors in both the static and dynamic environments. It could be observed that the inference by the CNN in the static environment is sometimes accurate but usually noisy. Additionally, the inference in the dynamic environment is significantly noisy.

TABLE I

ESTIMATION ERRORS OF E2E LOCALIZATION IN THE CAR-GARAGE ENVIRONMENT.

	Static		Dynamic	
	Pos. [m]	Ori. [deg]	Pos. [m]	Ori. [deg]
Ave.	0.693	2.596	2.800	21.412
Std.	0.469	2.361	2.757	33.821
Min.	0.031	0.009	0.131	0.007
Max.	2.701	22.954	20.472	176.400

TABLE II

ESTIMATION ERRORS WITHOUT WHEEL-SLIPPAGE SIMULATION IN STATIC OUTDOOR ENVIRONMENT. THE POSITION AND ORIENTATION ERRORS ARE SHOWN IN METER AND DEGREE, RESPECTIVELY.

	Proposed method		MCL		E2E	
	Pos.	Ori.	Pos.	Ori.	Pos.	Ori.
Ave.	0.323	0.929	0.307	0.892	1.765	3.629
Std.	0.152	0.861	0.148	0.828	1.600	3.880
Min.	0.001	0.000	0.005	0.000	0.003	0.000
Max.	0.889	5.788	0.819	6.487	27.911	58.480

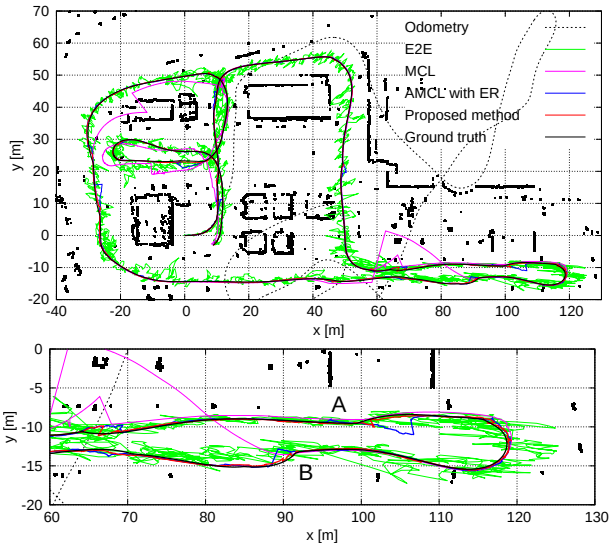

Fig. 3. Localization results in the outdoor environment.

Through the pre-experiments, we observed that E2E localization is not adequate for precise autonomous navigation even in static environments. In the experiments described below, we focus on static environments to validate the improvements in the estimation accuracy using the proposed fusion method. Notably, the CNN is re-trained using a training dataset created in the target environments beforehand.

B. Simulation experiments without wheel slippages

We first validated the localization performances of the proposed method, MCL, and E2E localization without wheel-slippage simulation in the simulated static outdoor environment. The result of E2E localization is the average pose of the learning-based particles. Table II lists the estimation errors using the three methods. MCL accurately estimated the robot pose because the experimental conditions were ideal. However, E2E localization was unstable.

The average localization accuracy of the proposed method was slightly lower than that of MCL because the proposed method fuses the learning-based particles to localize the pose, as described in equation (16). Although the E2E localization drastically failed, the localization performance was not degraded. This result demonstrated that the performance of the proposed method is almost the same as that of MCL working in the ideal environment.

C. Simulation experiments with wheel slippages

We then validated the localization performances using simulations with wheel slippages. There was a 1 % chance of wheel slippage, and the slippage occurred according to the white noise with a variance of $(15\ \mathrm{deg})^2$.

In the experiments, we also used augmented MCL (AMCL) [5] with ER [16] as the comparison method. AMCL observes the history of the likelihood to detect the mis-convergence, i.e., the failures estimated by the MCL[1]. ER is the method used to recover from the failure estimates. It is suitable for recovering the position tracking failures because the particle distribution is expanded around the current estimated pose if localization failure is detected.

Figure 3 shows the estimated trajectories in the outdoor environment. Although MCL handled the wheel slippages when the slippage was small, it failed the estimate when the angular disturbance was large, e.g., exceeding 30 degrees. The E2E localization was also unstable in this experiment.

The proposed method and AMCL with ER successfully tracked the robot pose. The bottom part of Fig. 3 shows an enlarged view of the top part. Even upon the occurrence of wheel slippages at A and B, both the methods recovered from the failure estimates. The proposed method achieved the quick recover more than AMCL. Because AMCL needs to observe the likelihood history, several estimation steps are necessary to detect the failures. In contrast, the proposed method does not need to detect the failure because the E2E localization is always fused with MCL. Consequently, the proposed method could achieve quick recovery and its estimation accuracy was higher than that of AMCL.

We also conducted experiments in an indoor environment with a long corridor and obtained the same result as that obtained from the outdoor environment experiment, i.e., the proposed method and AMCL with ER accomplished pose tracking, and the estimation accuracy of the proposed method was the highest. Table III lists the estimation errors of the proposed method, AMCL with ER, and E2E localization in both the experiments. Note that the target path and environment to obtain the results shown in Tables II and III are the same. Because MCL could not succeed in pose tracking, its results were not included in the table. We confirmed that the proposed method can robustly estimate the robot pose despite

[1]More specifically, the AMCL calculates the random particle rate that should be injected in the re-sampling phase to recover from the mis-convergence while observing the likelihood history [5]. The estimate is considered to be unsuccessful if the random particle rate exceeds zero.

TABLE III

POSE-ESTIMATION ERRORS BY THE PROPOSED METHOD, AMCL WITH ER, AND E2E LOCALIZATION WITH WHEEL-SLIPPAGE SIMULATION.

	Outdoor shown in Fig. 3						Indoor shown in Fig. 5					
	Proposed method		AMCL with ER		E2E		Proposed method		AMCL with ER		E2E	
	Pos.	Ori.	Pos.	Ori.	Pos.	Ori.	Pos.	Ori.	Pos.	Ori.	Pos.	Ori.
Ave.	0.343	1.486	0.383	2.710	1.850	4.078	0.200	2.208	0.244	4.101	1.542	6.517
Std.	0.160	2.466	0.262	5.073	1.614	4.551	0.133	2.954	0.221	6.160	2.703	18.252
Min.	0.002	0.002	0.001	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.003	0.000
Max.	1.194	28.587	2.465	36.699	24.852	58.114	0.890	42.170	1.545	46.162	20.511	162.511

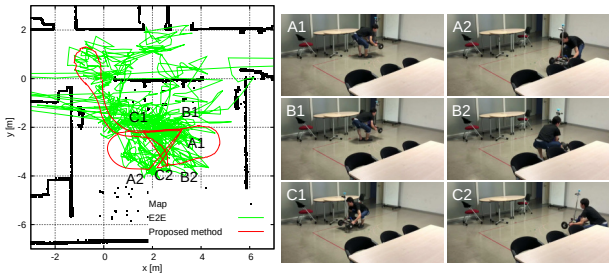

Fig. 4. Experiment with the real robot. The robot was kidnapped at A, B, and C from 1 to 2. The accompanying video shows this experiment.

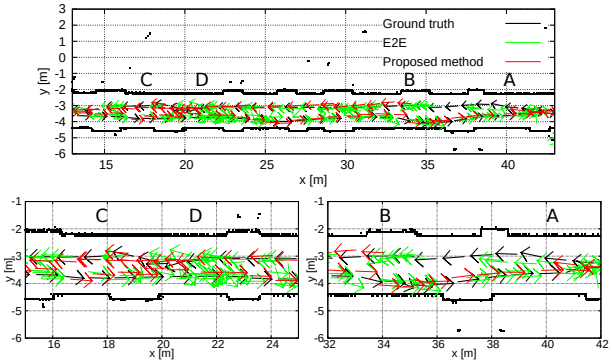

Fig. 5. Failure case examples of pose estimate using the proposed method. The bottom figures are the enlarged views of the top figure.

the presence of unanticipated errors. However, we found limitation in the proposed method through the experiment. The limitations will be detailed in Section V-E.

D. Real experiments

We also conducted experiments in a real indoor environment. The robot was equipped with a Hokuyo 2D LiDAR (UTM-30LX) and wheel encoders. The training dataset for E2E localization was created using the simulation. Notably, the real-environment experiments only demonstrated the qualitative results because the ground truth was not available.

Figure 4 shows the trajectories estimated by the proposed method and E2E localization with unanticipated errors. In this experiment, we kidnapped the robot three times at A, B, and C from 1 to 2, where *kidnapping* means that the localization suddenly fails because the robot jumps. However, the proposed method could quickly recover from the localization failures owing to the integration with E2E localization. Furthermore, smooth trajectory estimation could be achieved even though the E2E localization was not precise. From the real-environment experiments, we confirmed that the proposed method also performed well in the real environment, similar to the simulation experiments.

E. Limitation

We found the limitation of the proposed method through the experiment conducted in the indoor environment with the long corridor. Because the proposed method fuses the MCL and E2E localization, it fails to estimate the pose if both the methods fail their estimates. Figure 5 shows such an example. In the long corridor area, similar measurements were obtained and the learning-based particles were sometimes drawn into incorrect areas. If the MCL failed to estimate the pose in such a case, the proposed method also failed the pose estimate. At point A, wheel slippage occurred, and this problem was found. As a result, the pose estimate result using the proposed method jumped to the point B.

Although the learning-based particles were correctly sampled after the jump, the proposed method could not immediately re-localize the robot pose because the likelihood of the learning-based particles is calculated using the predictive distribution. However, the calculation of likelihood using the predictive distribution prevents the influence of unstable estimates by E2E localization. This results shows a trade-off problem between accuracy and robustness. When the robot arrived at point C, the proposed method correctly estimated the robot pose, indicated as point D, because the likelihood calculation using the measurement model yielded obvious differences. A more robust measurement model needs to be developed to solve this trade-off problem.

VI. CONCLUSION

This paper has presented a hybrid localization method that fuses the MCL and E2E localization via IS. By using the proposed fusion method, the advantages of both the localization methods can be simultaneously leveraged while mitigating their disadvantages. The experiments demonstrated that the proposed method can smoothly estimate the robot pose, similar to the model-based method, and quickly re-localize it from the failures, similar to the learning-based method.

Our future work aims to develop as E2E localization method that is robust toward environmental changes and an automatic dataset creation method for E2E localization while utilizing our previous work [41]–[43].

ACKNOWLEDGMENT

This work was supported by the JST (Nagoya-COI) and KAKENHI under grant number 40786092.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328, 1999.

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

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[7] P. Maybeck. Stochastics models, estimation, and control. *Academic Press*, 1, 1979.

[8] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems*, pages 563–569, 2000.

[9] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv:1506.02158*, 2015.

[10] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE International Conference on Robotics and Automation*, pages 4762–4769, 2016.

[11] A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584. 2017.

[12] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the National Conference on Artificial Intelligence*, pages 343–349, 1999.

[13] J. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proceedingsof the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 736–743, 1998.

[14] S. Thrun, D. Fox, W. Burgard, and F. Dallaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1–2):99–141, 2001.

[15] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1225–1232, 2000.

[16] R. Ueda, T. Arai, K. Sakamoto, T. Kikuchi, and S. Kamiya. Expansion resetting for recovery from fatal error in Monte Carlo localization - Comparison with sensor resetting methods-. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2481–2486, 2004.

[17] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

[18] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2017.

[19] R.Q. Charles, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[20] J. Li, H. Zhan, B. M. Chen, I. Reid, and G. H. Lee. Deep learning for 2D scan matching and loop closure. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 763–768, 2017.

[21] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018.

[22] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6939–6946, 2018.

[23] N. Radwan, A. Valada, and W. Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *arXive:1804.08366*, 2018.

[24] D. Cattaneo, M. Vaghi, A. L. Ballardini, S. Fontana, D. G. Sorrenti, and W. Burgard. CMRNet: Camera to LiDAR-map registration. *arXiv:1906.10109*, 2019.

[25] W. Yuan, D. Held, C. Mertz, and M. Hebert. Iterative transformer network for 3D point cloud. *arXiv:1811.11209*, 2018.

[26] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li. LO-Net: Deep real-time lidar odometry. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 8473–8482, 2019.

[27] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song. L^3-Net: Towards learning based LiDAR localization for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[28] A. Amini, G. Rosman, S. Karaman, and D. Rus. Variational end-to-end navigation and localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019.

[29] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf. Dropout sampling for robust object detection in open-set conditions. 2017.

[30] V. Peretroukhin, L. Clement, and J. Kelly. Reducing drift in visual odometry by inferring sun direction using a Bayesian convolutional neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2035–2042, 2017.

[31] S. Wang, R. Clark, H. Wen, and N. Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5):513–542, 2017.

[32] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *European Conference on Computer Vision*, 2018.

[33] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé. Understanding the limitations of CNN-based absolute camera pose regression. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3302–3312, 2019.

[34] 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.

[35] 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.

[36] S. Nobili, G. Tinchev, and M. Fallon. Predicting alignment risk to prevent localization failure. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1003–1010, 2018.

[37] H. Almqvist, M. Magnusson, T. P. Kucner, and A. J. Lilienthal. Learning to detect misaligned point clouds. *Journal of Field Robotics*, 35(5):662–677, 2018.

[38] A. Krizhevsky, I. Sutskever, and G. E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[39] F. Chollet et al. Keras. `https://keras.io`, 2015.

[40] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transaction on Robotics*, 23(1):34–46, 2007.

[41] 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.

[42] 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.

[43] 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.