Reliable Monte Carlo Localization for Mobile Robots

Naoki Akai[1,2*]
[1]Graduate School of Engineering, Nagoya University
Furo, Chikusa, Nagoya, Japan
[2]LOCT Co., Ltd.
Ikatsu, Showa, Nagoya, Japan
`akai@nagoya-u.jp`

Abstract

Reliability is a key factor for realizing safety guarantee of fully autonomous robot systems. In this paper, we focus on reliability in mobile robot localization. Monte Carlo localization (MCL) is widely used for mobile robot localization. However, it is still difficult to guarantee its safety because there are no methods determining reliability for MCL estimate. This paper presents a novel localization framework that enables robust localization, reliability estimation, and quick re-localization, simultaneously. The presented method can be implemented using a similar estimation manner to that of MCL. The method can increase localization robustness to environment changes by estimating known and unknown obstacles while performing localization; however, localization failure of course occurs by unanticipated errors. The method also includes a reliability estimation function that enables a robot to know whether localization has failed. Additionally, the method can seamlessly integrate a global localization method via importance sampling. Consequently, quick re-localization from a failure state can be realized while mitigating noisy influence of global localization. We conduct three types of experiments using wheeled mobile robots equipped with a 2D LiDAR. Results show that reliable MCL that performs robust localization, self-failure detection, and quick failure recovery can be realized.

1   Introduction

Reliability is a key factor for realizing safety guarantee of fully autonomous robot systems. Localization is a fundamental module for autonomous navigation systems. In this work, we focus on 2D LiDAR-based localization and present a method that enables to make localization more reliable. To realize reliable localization, we consider that localization must

- robustly work in dynamic environments;

- immediately detect failure of its estimate;

- quickly re-localize the robot pose if failed, i.e., quick failure recovery.

The method presented in this work can achieve these functions simultaneously. It should be noted that we only present 2D-LiDAR-based implementation of the presented framework in this paper, but the framework can be applied to other localization problems such as 3D-LiDAR-based one.

---

*Home page of the corresponding author: https://sites.google.com/view/naokiakaigoo/home

The presented method is composed of two processes; pose tracking and global localization. The pose tracking process is basically implemented based on Monte Carlo localization (MCL) [Thrun et al.(2005)Thrun, Burgard, and Fox]. Additionally, the sensor measurement class and reliability estimation methods we previously presented in [Akai et al.(2018a)Akai, Morales, and Murase, Akai et al.(2018b)Akai, Morales, and Murase] are integrated. The global localization process is implemented based on the free-space feature presented in [Millane et al.(2019)Millane, Oleynikova, Nieto, Siegwart, and Cadena]. Global localization is necessary to achieve re-localization from failure of pose tracking; however, global localization does not stably work more than pose tracking in usual. To seamlessly fuse global localization and pose tracking, the probabilistic fusion method we previously presented in [Akai et al.(2020)Akai, Hirayama, and Murase] is employed. By using these methods, the above functions can be achieved.

The contribution of this paper is achieving integration of the above mentioned methods. As mentioned above, the presented method is basically composed of our previous proposals. However, we did not achieve these integration owing to less of a model and implementation difficulty. In this work, we present a new graphical model for the pose tracking process and solve the global localization problem using the free-space feature. As a result, reliable localization that simultaneously achieves the above three things can be realized. We conduct simulation-, dataset- and our-own-platform-based experiments using wheeled mobile robots equipped with a 2D LiDAR. Through the simulation-based experiments, we show that the presented method numerically works well more than traditional methods. Through the dataset- and our-own-platform-based experiments, we show that the presented method also works with actual robots. The software used in this work is publicly available at[1]. The contribution is summarized as follows.

- Presenting a new framework that integrates our previous proposals presented in [Akai et al.(2018a)Akai, Morales, and Murase, Akai et al.(2018b)Akai, Morales, and Murase, Akai et al.(2020)Akai, Hirayama, and Murase] and achieves reliable localization that simultaneously performs robust localization, reliability estimation, and quick re-localization

- Publishing ROS-based implementation of the new framework with 2D LiDAR as open source software

The rest of this paper is organized as follows. Section 2 summarizes related works. Section 3 describes the problem setting and details the presented method. Section 4 details implementation of the presented method. Section 5 and 6 describe experimental results using simulation, dataset and our own platform. Section 7 concludes this work.

2   Related work

This section summarizes existing works related to robust localization, reliability estimation, and re-localization.

2.1   Robustness

To perform localization, we need to model how sensor measurements are obtained. This model is referred to the measurement model [Thrun et al.(2005)Thrun, Burgard, and Fox]. The traditional measurement models are the beam and likelihood field models presented in [Thrun et al.(2005)Thrun, Burgard, and Fox]. These models consider that unknown obstacles, that is, obstacles do not exist on the map, are measured. Hence, these models enable to perform localization in dynamic environments. However, it is difficult for these models to work in highly dynamic environments because highly dynamic environment yields inconsistency of the models.

---

[1]`https://github.com/NaokiAkai/als_ros`

The major dynamics in real world are moving obstacles such as pedestrians, bicycles, and cars. The simplest method to cope with them is to treat them as outliers [Fox et al.(1999)Fox, Burgard, and Thrun,Burgard et al.(1999)Burgard, Cremers, Fox, Hahnel, Lakemeyer, Schulz, Steiner, and Thrun,Montemerlo et al.(2002)Montemerlo, Thrun, and Whittaker,Anguelov et al.(2002)Anguelov, Biswas, Koller, Limketkai, Sanner, and Thrun,Hähnel et al.(2003)Hähnel, Triebel, Burgard, and Thrun,Schulz et al.(2003)Schulz, Fox, and Hightower]. In other words, measurements obtained from these obstacles are ignored in the likelihood calculation process. This approach effectively works for such dynamic obstacles; however, it cannot cope with semi-dynamic obstacles such as parked cars and removal of mapped obstacles.

An other effective approach to cope with dynamic obstacles is to model static and dynamic parts of the environment. Wolf and Sukhatme [Wolf and Sukhatme(2005)] proposed an occupancy grid mapping method that separately maps static and dynamic parts. Montesano *et al.* [Montesano et al.(2003)Montesano, Minguez, and Montano] also proposed a modeling method of static and dynamic parts. This modeling is achieved using a set of filters tracking the moving objects and a map of the static structure constructed online. Dynamic environment modeling methods have been extended by many authors [Brechtel et al.(2010)Brechtel, Gindele, and Dillmann,Meyer-Delius et al.(2012)Meyer-Delius, Beinhofer, and Burgard,Saarinen et al.(2012)Saarinen, Andreasson, and Lilienthal,Tipaldi et al.(2013)Tipaldi, Meyer-Delius, and Burgard]. Wang *et al.* [Wang et al.(2007)Wang, Thorpe, Thrun, Hebert, and Durrant-Whyte] extended the probabilistic method and proposed a general framework for simultaneous localization, mapping, and dynamic object tracking. Biber and Duckett [Biber and Duckett(2005)] proposed a modeling method that represent multiple timescales to select the sensor measurements of the most appropriate timescale. Meyer-Delius *et al.* [Meyer-Delius et al.(2010)Meyer-Delius, Hess, Grisetti, and Burgard] and Valencia *et al.* [Valencia et al.(2014)Valencia, Saarinen, Andreasson, Vallvé, Andrade-Cetto, and Lilienthal] also proposed a localization method that uses temporal maps. These approaches are able to provide a better static map for localization. However, these approaches require increasing of memory cost for modeling. The presented method increases localization robustness without increasing memory and computational costs.

Improving the measurement model is also effective to improve localization robustness. Olufs and Vincze [Olufs and Vincze(2009)] and Takeuchi *et al.* [Takeuchi et al.(2010)Takeuchi, Ohno, and Tadokoro] proposed the measurement model that utilizes the free space. Appropriate sensor measurement selection such as presented by Kim and Chung [Kim and Chung(2018)] is also effective to improve robustness. However, these extensions are not to extend the graphical model for localization presented in [Thrun et al.(2005)Thrun, Burgard, and Fox]. Our approach extends the graphical model and formulates the simultaneous localization and sensor measurement class estimation problem (see Fig. 1). The basic model used in our approach is different from that of used in these approaches. Consequently, localization robustness to environment changes can be improved.

Our approach to improve localization robustness is similar to the method presented by Yang and Wang [Yang and Wang(2011)]. In their method, the feasibility grids that maintain the stochastic estimates of the feasibility (crossability) states of the environment are used. Based on the feasibility grids, sensor measurements can be decomposed into stationary and moving objects. Our approaches also uses two class conditional measurement models for known and unknown obstacles; however, no other information is used to calculate them such as the feasibility grids.

2.2 Reliability

MCL is a well-known algorithm for mobile robot localization [Dellaert et al.(1999)Dellaert, Fox, Burgard, and Thrun.] and is based on Bayesian filtering. MCL does not include a failure detection function in its standard implementation since the Bayesian filter just estimates the posterior over the target variable, i.e., robot pose. Gutmann and Fox [Gutmann and Fox(2002)] proposed a failure detection method for MCL that observes history of likelihood. MCL including this method is referred to augmented MCL (AMCL). Failure detection by AMCL is performed based on thresholds.

Scan registration methods such as iterative closest points (ICP) [Besl and McKay(1992)] and normal distributions transform (NDT) [Biber and Straßer(2003)] are also widely used for mobile robot localization. These methods also do not have an exact failure detection function in its standard implementation. The simplest method to detect failures for these algorithms is setting thresholds for distances between corresponding points and/or features [Rusinkiewicz and Levoy(2001)]. However, insufficient performance of the threshold-based method when registration errors are small is presented in [Silva et al.(2005)Silva, Bellon, and Boyer]. In the presented method, the threshold-based method is also used; however, we can improve classification stability because the proposed graphical model derives the Bayesian filter for reliability estimation.

Quddus *et al.* [Quddus et al.(2006)Quddus, Ochieng, and Noland] presented an empirical methods that use map matching to estimate localization integrity. In this method, fuzzy logic is used to determine a metric between 0 and 100 that represents uncertainty (confidence) of the map matching results. However, we consider that confidence and reliability for localization results are different because confidence can be calculated from the pose estimation uncertainty, but reliability cannot be calculated. Our approach employees a localization classifier and estimates reliability using the classifier.

Sundvall and Jensfelt [Sundvall and Jensfelt(2006)] and Mendoza *et al.* [Mendoza et al.(2018)Mendoza, Veloso, and Simmons] proposed a localization failure detection method using a redundant positioning system. The main idea of these methods is to use majority vote of the redundant system to detect failures. In other words, these methods do not estimate correctness of each estimate in the redundant system. Our approach explicitly estimates reliability of the localization system. Al Hage *et al.* [Al Hage et al.(2022)Al Hage, Xu, Bonnifait, and Ibanez-Guzman] proposed a method for addressing localization integrity that combines measurement rejection and position error characterization. In this method, a multi-sensor data fusion with a fault detection and exclusion algorithm is constituted using a bank of information filters. Our approach does not use multi-sensor data for estimating reliability.

Modeling localization failure is not trivial. Hence, machine learning approaches have been applied recently. Almqvist *et al.* [Almqvist et al.(2018)Almqvist, Magnusson, Kucner, and Lilienthal] applied several threshold- and machine-learning-based methods to classify misaligned point clouds. Alsayed *et al.* [Alsayed et al.(2017)Alsayed, Bresson, A, and Nashashibi, Alsayed et al.(2018)Alsayed, Bresson, A, and Nashashibi] presented a machine-learning-based failure detection method for 2D LiDAR SLAM. Similar approaches for GNSS-based localization is also presented in [Hsu(2017)]. Zhen *et al.* [Zhen et al.(2017)Zhen, Zeng, and Scherer] presented "localizability" that represents possibility whether localization works or not in each area. The localizability cannot exactly describe localization performance because it is determined using only geometric shape of the map, i.e., it does not consider on-line situation. Nobili *et al.* [Nobili et al.(2018)Nobili, Tinchev, and Fallon] extended the localizability and proposed "alignability" that represents alignment risk. In addition, support vector machine is applied to learn relationship between the alignability and localization failures. These methods can indeed detect localization failures; however, these methods cannot consider uncertainty of the classifier. Our approach can consider the uncertainty because reliability estimation is performed based on the Bayesian filtering using the classification results.

It should be noted that the estimated reliability by the presented method cannot perfectly guarantee localization correctness since it is also estimated without the ground truth. Interpretation of reliability in this work is discussed in Section 6.3.2.

2.3 Global localization and re-localization

MCL can be also used for global localization [Dellaert et al.(1999)Dellaert, Fox, Burgard, and Thrun.]. However, standard MCL is not suitable for solving the global localization problem. To efficiently solve the global localization problem using MCL, there are several extensions. It should be noted that we treat that the global localization and re-localization problems are the same because these problems aim to estimate the current pose without accurate initial and/or previous estimates.

Lenser and Veloso [Lenser and Veloso(2000)] proposed the sensor resettings that enables to perform re-localization. In the sensor resettings, failure of MCL estimate is detected with the scheme used in AMCL. If failure is detected, the particles are sampled using the measurement model. Then, all the particles are evaluated using the measurement model and perform re-localization. Ueda *et al.* [Ueda et al.(2004)Ueda, Arai, Sakamoto, Kikuchi, and Kamiya] proposed the expansion resettings that also enables to perform re-localization. In the expansion resettings, failure of MCL estimate is also detected with the AMCL's scheme. If failure is detected, the particle distribution is expanded around the current estimate. The expansion resettings is suitable for recovering small estimation errors caused by such as wheel slippage and collision against obstacles.

Thrun *et al.* [Thrun et al.(2001a)Thrun, Fox, Burgard, and Dallaert] proposed mixture MCL (MMCL) that uses multiple proposal distributions and importance sampling to fuse particles generated from multiple distributions. In [Thrun et al.(2001a)Thrun, Fox, Burgard, and Dallaert], the measurement model is used as the proposal distribution. Hence, this method is similar to the sensor resettings; however, it differs from the sensor resettings because the fusion is performed based on the importance sampling. However, accurate sampling using the measurement model is not a trivial task in an era when MMCL has been proposed because of the computational and memory costs. Thrun *et al.* [Thrun et al.(2001b)Thrun, Langford, and Verma] also proposed a risk sensitive particle filter (RSPF) that considers the risk estimated by Markov decision process. In [Thrun et al.(2001b)Thrun, Langford, and Verma], it was shown that RSPF achieved faster recovery from localization failure more than standard MCL.

Recent evolution of neural networks also contributes to the localization filed. Kendall *et al.* [Kendall et al.(2015)Kendall, Grimes, and Cipolla] proposed PoseNet that directly estimates 6-DoF pose from a camera image, i.e., networks take sensor measurements and infer location where the sensor measurements are obtained and such networks are referred to end-to-end (E2E) networks. Gal and Ghahramani [Gal and Ghahramani(2015)] showed that output of neural networks can be treated as posterior by approximating using variational inference. Kendall and Cipolla [Kendall and Cipolla(2016)] used this idea and showed that uncertainty of E2E-based localization can be estimated. This means that poses can be sampled from the probability over the measurement, i.e., this sampling is similar to sampling from the measurement model. Sun *et al.* [Sun et al.(2020)Sun, Adolfsson, Magnusson, Andreasson, Posner, and Duckett] and we [Akai et al.(2020)Akai, Hirayama, and Murase] used this sampling method and proposed a localization method that fuses MCL and E2E-based localization. Owing to the fusion, smooth pose tracking and fast re-localization can be simultaneously achieved.

However, the use of E2E networks requires users preparation of training dataset because coordinates of a map depend on how to build a map. In other words, E2E networks are needed to be trained and this training depends on environment and map building process. We consider that this requirement is not suitable for the use in open source software. Hence, we use a model-based method presented in [Millane et al.(2019)Millane, Oleynikova, Nieto, Siegwart, and Cadena]. This re-localization performance might not overcome the E2E-based one; however, we show that accurate and fast re-localization can be achieved using the fusion based on the importance sampling. It should be noted that the opened software is implemented to easily change the global localization method, that is, readers can easily change it according to application, for example, GNSS and AR marker.

3   Proposed method

This section first describes the problem setting. Then, the pose tracking and global localization methods contained in the presented localization framework are detailed.
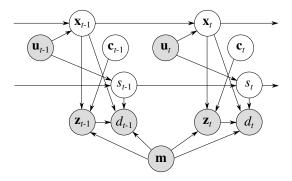
Figure 1: The graphical model for the pose tracking.

3.1 Problem setting

In this work, we focus on the 2D localization problem for mobile robots. A pose of a target robot is composed of 2D position, x and y, and heading angle, θ. We assume that a robot is equipped with a 2D LiDAR and an inertial navigation system (INS). Measurements of INS are denoted as $\mathbf{u}$ and are used to predict the robot pose. Measurements of LiDAR are denoted as $\mathbf{z}$ and are used to perform matching with a map denoted as $\mathbf{m}$. In this work, we aim to estimate the current robot pose with sequences of $\mathbf{u}$ and $\mathbf{z}$, and $\mathbf{m}$. Additionally, we try to realize robust localization, reliability estimation, and quick re-localization simultaneously. Robust localization and reliability estimation are realized by the improved pose tracking algorithm described in the next subsection. Quick re-localization is realized by the combination of global localization and the probabilistic fusion of it with the pose tracking algorithm.

3.2 Graphical model for pose tracking

3.2.1 Variable definition

Figure 1 illustrates the graphical model for the pose tracking process. The white and gray nodes indicate hidden and observable variables. The robot pose $\mathbf{x}$, measurement classes $\mathbf{c}$, and localization state s are treated as the hidden variables. The INS and LiDAR measurements, $\mathbf{u}$ and $\mathbf{z}$, map $\mathbf{m}$, and output of a localization state classifier d are treated as the observable variables.

The LiDAR measurements are denoted as $\mathbf{z} = (\mathbf{z}^{[1]}, \mathbf{z}^{[2]}, ..., \mathbf{z}^{[K]})$, where $\mathbf{z}^{[k]}$ denotes kth measurement. The measurement classes $\mathbf{c}$ indicate category of the LiDAR measurements and denoted as $\mathbf{c} = (c^{[1]}, c^{[2]}, ..., c^{[K]})$, where $c^{[k]}$ corresponds to $\mathbf{z}^{[k]}$. In this work, we use two classes denoted as $c^{[k]} \in \mathcal{C} = \{\text{known}, \text{unknown}\}$. These classes indicate whether the measurement is obtained from obstacles existing on the map or not.

The localization state classifier distinguishes whether localization has failed. The classifier can be implemented using any methods such as threshold- and machine-learning-based methods [Almqvist et al.(2018)Almqvist, Magnusson, Kucner, and Lilienthal]. Output form of the classifier is changed according to its implementation. In this work, we use a threshold-based classifier that sets a threshold to mean absolute error (MAE) defined using residual errors and d is to be an one dimensional continuous value, where the residual errors represent a distance set between LiDAR measurement points and the closest mapped obstacle for each point. The localization state s is denoted as $s \in \mathcal{S} = \{\text{success}, \text{failure}\}$. These states indicate whether localization has succeeded. The state s is estimated using outputs of the classifier d. Since $s = \text{success}$ indicate that localization has succeeded, reliability can be known by calculating its probability, i.e., $p(s = \text{success})$.

3.2.2 Formulation

Our objective is to estimate the joint posterior distribution over the robot pose, measurement classes, and localization state at current time t as shown in Eq. (1).

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

where $1 : t$ indicates time sequence, i.e., $\mathbf{u}_{1:t} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_t)$. Eq. (1) can be decomposed using the multiplication theorem

$$
\begin{aligned}
p(\mathbf{x}_t, \mathbf{c}_t, s_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) = \quad & p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) && (2) \\
& p(\mathbf{c}_t | \mathbf{x}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) && (3) \\
& p(s_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}), && (4)
\end{aligned}
$$

where Eqs. (2), (3), and (4) denote the posterior over the robot pose, measurement classes, and localization state, respectively.

Eqs. (2), (3), and (4) cannot be calculated as it is. We formulate, respectively, them to enable to calculate. The Bayes theorem is first applied to Eq. (2).

$$
\begin{aligned}
&p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) \\
&= \frac{p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) p(d_t | \mathbf{x}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m}) p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m})}{p(\mathbf{z}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) p(d_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m})} \\
&= \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) p(d_t | \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m})
\end{aligned}
$$

To change from the line 2 to 3, the denominator was re-written as a normalization constant, η, and D-separation [Bishop(2006)] was applied to the distributions over $\mathbf{z}_t$ and d_t to remove non related conditional variables. The law of total probability is then applied to each distribution.

$$
\begin{aligned}
&p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) p(d_t | \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) \\
&= \sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m}) p(\mathbf{c}_t) \sum_{s_t \in \mathcal{S}} p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m}) p(s_t) \\
&\qquad \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) \mathrm{d}\mathbf{x}_{t-1} \\
&= \sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m}) p(\mathbf{c}_t) \sum_{s_t \in \mathcal{S}} p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m}) p(s_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) \mathrm{d}\mathbf{x}_{t-1}
\end{aligned}
$$

To change from the line 2 to 3, D-separation was applied to the distribution over $\mathbf{x}_t$ and $\mathbf{u}_t$ was omitted from the distribution over $\mathbf{x}_{t-1}$ since the future input does not affect to the previous pose. The recursive update equation regarding the pose distribution is obtained.

Eq. (3) can be formulated by applying the Bayes theorem and D-separation.

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

To formulate Eq. (4), the Bayes theorem and D-separation are first applied.

$$
\begin{aligned}
p(s_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) &= \eta p(d_t | \mathbf{x}_t, \mathbf{c}_t, s_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m}) p(s_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m}) \\
&= \eta p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m}) p(s_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m})
\end{aligned}
$$

Then, the law of total probability is applied to the second distribution.

$$
\begin{aligned}
&p(s_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m}) \\
&= \sum_{s_{t-1} \in \mathcal{S}} p(s_t | \mathbf{x}_t, \mathbf{c}_t, s_{t-1}, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m}) p(s_{t-1} | \mathbf{x}_t, \mathbf{c}_t, \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t-1}, \mathbf{m}) \\
&= \sum_{s_{t-1} \in \mathcal{S}} p(s_t | s_{t-1}, \mathbf{u}_t) p(s_{t-1} | \mathbf{x}_{t-1}, \mathbf{c}_{t-1}, \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m})
\end{aligned}
$$

To change from the line 2 to 3, D-separation was applied to the distribution over s_t and $\mathbf{x}_t$, $\mathbf{c}_t$, $\mathbf{u}_t$, and $\mathbf{z}_t$ was omitted from the distribution over s_{t-1} since the future conditions do not affect to the previous state. The recursive update equation regarding the localization state is also obtained.

Finally, the target distribution is denoted as shown in Eq. (5).

$$
\begin{aligned}
&p(\mathbf{x}_t, \mathbf{c}_t, s_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m}) \\
&= \eta \sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m}) p(\mathbf{c}_t) \sum_{s_t \in \mathcal{S}} p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m}) p(s_t) \underbrace{\int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) \mathrm{d}\mathbf{x}_{t-1}}_{\text{robot pose}} \\
&\underbrace{p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m}) p(\mathbf{c}_t)}_{\text{measurement classes}} \\
&\underbrace{p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m}) \sum_{s_{t-1} \in \mathcal{S}} p(s_t | s_{t-1}, \mathbf{u}_t) p(s_{t-1} | \mathbf{x}_{t-1}, \mathbf{c}_{t-1}, \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m})}_{\text{localization state}},
\end{aligned}
$$

(5)

In Eq. (5), we have four important models denoted as $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m})$, $p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m})$, $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$, and $p(s_t | s_{t-1}, \mathbf{u}_t)$. These models are referred to the class conditional measurement, decision, motion, and reliability transition models, respectively. These models are detailed in Section 4.2, 4.3, 4.5.1, and 4.5.2, respectively.

In this work, we use Rao-Blackwellized particle filter (RBPF) to estimate Eq. (5). More precisely, Eq. (2) is estimated using particle filter (PF) and Eqs. (3) and (4) are estimated using analytical methods based on sampled poses in PF.

3.3 Global localization and its fusion with pose tracking

In this work, we assume that the global localization problem is to estimate the probabilistic distribution over the robot pose under a condition where several INS and LiDAR measurements and map are given. Hence, we also try to estimate the distribution shown in Eq. (6).

$$
p(\mathbf{x}_t | \mathbf{u}_{t-N+1:t}, \mathbf{z}_{t-N:t}, \mathbf{m}),
$$

(6)

where N is the number of previous steps used for global localization. It should be noted that the INS measurement $\mathbf{u}$ is ignored if $N = 0$.

As shown in Eqs. (2) and (6), there are two probabilistic distributions over the robot pose in the presented method. Ideally, global localization has to be performed only when localization has failed. However, exact failure detection of localization is challenging owing to the use of the independent assumption to the LiDAR measurements [Thrun et al.(2005)Thrun, Burgard, and Fox, Akai et al.(2019)Akai, Morales, Hirayama, and Murase, Akai et al.(2022)Akai, Akagi, Hirayama, Morikawa, and Murase]. Even though the presented method estimates reliability of the localization result, it is difficult to perfectly classify whether localization has failed using the estimated reliability. Namely, seamless fusion of pose tracking and global localization is preferable more than switching of them according to estimate condition.

To realize the seamless fusion, we use the importance sampling. First, poses are sampled from Eq. (6). Let $^{\mathrm{G}}M$ be a number of sampled poses and $^{\mathrm{G}}\mathbf{x}_t^{[i]}$ be a ith sampled pose, we assume that the sampled poses approximate Eq. (6) like the approximation by PF.

$$
p(\mathbf{x}_t | \mathbf{u}_{t-N+1:t}, \mathbf{z}_{t-N:t}, \mathbf{m}) \simeq \frac{1}{^{\mathrm{G}}M} \sum_{i=1}^{^{\mathrm{G}}M} \delta(\mathbf{x}_t - {}^{\mathrm{G}}\mathbf{x}_t^{[i]}),
$$

(7)

where $\delta(\cdot)$ is the Dirac delta that is 1 if value within the blackett is 0, and 0 otherwise. By the approximation shown in Eq. (7), we can use the distribution shown in Eq. (6) as a proposal distribution, i.e., distribution for sampling the particles.

In PF, likelihood of the particles is determined by quotient of the target and proposal distributions. The target distribution used in the presented method is shown in Eq. (2). Hence, the likelihood of the particles sampled from Eq. (6) is denoted as

$$
\begin{aligned}
{}^{\mathrm{G}}\omega_t =& \frac{p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m})}{p(\mathbf{x}_t | \mathbf{u}_{t-N+1:t}, \mathbf{z}_{t-N:t}, \mathbf{m})} \\
\simeq& \eta \frac{\sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m}) p(\mathbf{c}_t) \sum_{s_t \in \mathcal{S}} p(d_t | \mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m}) p(s_t)}{\frac{1}{{}^{\mathrm{G}}M} \sum_{i=1}^{{}^{\mathrm{G}}M} \delta(\mathbf{x}_t - {}^{\mathrm{G}}\mathbf{x}_t^{[i]})} \\
& \cdot \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) \mathrm{d}\mathbf{x}_{t-1},
\end{aligned}
\tag{8}
$$

where $\int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m}) \mathrm{d}\mathbf{x}_{t-1}$ is referred to predictive distribution.

In the robot pose distribution shown in Eq. (5), the predictive distribution is used to sample the particles and their likelihood is calculated using the class conditional measurement and decision models because the predictive distribution is the proposal distribution. However, in the likelihood calculation for the sampled particles from Eq. (6), the predictive distribution is used. Owing to that, seamless fusion of pose tracking and global localization can be realized.

There are various methods to achieve sampling from Eq. (6). In this work, we use localization using the free-space feature presented in [Millane et al.(2019)Millane, Oleynikova, Nieto, Siegwart, and Cadena]. This localization method is detailed in Section 4.4.1.

4   Implementation

The system diagram implemented in this work to realize the functions described in Section 3 is illustrated in Fig. 2. The main modules are the pose tracker and global localizer. Both the modules receive moving velocities, scan, and occupancy grid map as input. The global localizer outputs sampled poses based on its estimate. The pose tracker receives the sampled poses and finally outputs estimated pose. In addition, the pose tracker outputs reliability to the localization result and measurement classes.

This system is composed of following processes.

1. initialization with a given initial pose

2. update the particles using the motion model

3. update reliability using the reliability transition model

4. perform global localization and sample the particles

5. calculate the likelihood of the particles updated by the motion model using the class conditional measurement and decision models

6. calculate the likelihood of the particles sampled from global localization using the class conditional measurement and decision models and the predictive distribution

7. estimate the robot pose, sensor measurement classes, and reliability
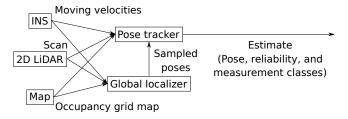
8. re-sampling the particles

Figure 2: The localization system diagram.

9. go back to 2 and repeat

These processes are detailed in this section.

4.1 Initialization

In this work, we focus on the 2D localization problem using a LiDAR. A state of the particle $\mathbf{s}_t$ contains 2D pose, x_t, y_t, and θ_t, and likelihood, ω_t. In addition, the particles have reliability denoted as $p(s_t = \text{success})$, where s_t is the localization state. The number of the particles to estimate Eq. (2) is constant and is denoted as $^{\mathrm{P}}M$. Poses of the particles are randomly sampled around a given initial pose in the initialization step. Reliability of all the particles is set to 0.5. In the presented method, we also have another particle set that is sampled from Eq. (6). The number of these particles is not constant and is denoted as $^{\mathrm{G}}M$.

As initialization for the map, we build a distance field (DF). Each cell of DF contains distance from the closest mapped obstacle. DF enables to efficiently calculate the class conditional measurement model. In addition, DF is used for global localization based on the free-space feature.

4.2 Motion model

We assume that the motion of the robot can be modeled using discrete-time state equation and is denoted as $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$, where $f(\cdot)$ is the discrete-time state equation. To update pose of the particles, random noise according to Gaussian is added to the INS measurement $\mathbf{u}_t$. Update of the particles using the motion model is denoted as

$$\mathbf{x}_t^{[i]} = f(\mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t^{[i]}), \quad \mathbf{u}_t^{[i]} \sim \mathcal{N}(\mathbf{u}_t, {}^{\mathrm{u}}\Sigma(\mathbf{u}_t)), \tag{9}$$

where $^{\mathrm{u}}\Sigma$ is a diagonal determined based on the INS measurements, that is, values of the diagonal are to be large (or small) if the INS measurements are large (or small). In the opened software, the differential drive and omni directional models are supported.

4.3 Reliability transition model

Basically, localization accuracy decreases according to move of the robot. Hence, we assume that reliability also decrease according to the move, i.e., $\hat{r}_t \leq r_{t-1}$, where $\hat{r}_t$ is updated reliability using the reliability transition model from the previous step. However, modeling the decrease of reliability is not trivial. We used a heuristical method to model the decrease as shown in Eq. (10).

$$\hat{r}_t = r_{t-1}(1 - \sum_i \alpha_i \Delta_i^2), \tag{10}$$

where α_i is a positive arbitrary constant and Δ_i is displacement between the time steps measured by INS, for example, translational and angular displacements, Δd and $\Delta\theta$, are used if the differential driving model is used.

In the implementation, we set all α_i are 0 because effect of the decision model in reliability update is major more than that of the reliability transition model. In addition, the decision-model-based update can be used in every estimation step. Hence, effect of the reliability transition model is minor if update cycle is fast, e.g., more than 10 Hz. These parameters must be appropriately tuned if the update cycle is slow. Owing to the reliability transition model, we can model that the localization result will be unreliable when sensor measurements are not applied to update its estimate long time.

4.4 Global localization

For global localization, we use the free-space feature presented in [Millane et al.(2019)Millane, Oleynikova, Nieto, Siegwart, and Cadena]. In this subsection, we briefly describe the free-space feature. For more details, please see the literature.

4.4.1 Free-space feature

Let $\mathbf{m}$ be an occupancy grid map (OGM) and is denoted as $\mathbf{m} = (m^{[1]}, m^{[2]}, ..., m^{[M]})$, where $m^i \in \{\text{occupied}, \text{free space}, \text{unknown}\}$ is a state of ith cell. DF is built based on OGM. A Gaussian filter is applied to DF and a Hessian matrix is calculated in each cell. Based on the Hessian matrices, maxima, minima, and saddles are detected and these are used as keypoints. It should be noted that the keypoints are only defined on the free space.

A rotational invariant feature is assigned to the keypoints. First, a dominant orientation of a keypoint is determined. A 36-bin gradient orientation histogram is built within an arbitrary size window. The direction of the maximum frequency value is determined as the dominant orientation of the keypoint. Then, a 17-bin histogram regarding relative gradient orientations to the dominant orientation is built. This histogram is assigned to the keypoint as a feature. In addition, a type of keypoint, i.e., maxima, minima, and saddles, and an average value of DF values in the window are assigned as a feature.

In the matching process of the keypoints, the types of keypoints and average DF values are first compared. If the types are the same and difference between the average DF values is less than a threshold, these keypoints are regarded as corresponding. If multiple correspondences are found, the sums of differences of the relative gradient orientations histograms are computed. If constant multiplication of the minimum sum is less than the second minimum sum, the keypoint with the minimum sum is regarded as corresponding.

4.4.2 Pose sampling

The free-space features described in Section 4.4.1 are defined to the global map $\mathbf{m}$ in advance. In the localization phase, the LiDAR measurements are accumulated based on the INS measurements and a local map $^{\mathrm{L}}\mathbf{m}$ is built. In other words, the local map is built on the odometry frame. The pose of the odometry frame is denoted as $^{\mathrm{O}}\mathbf{x}_t = (^{\mathrm{O}}x_t, {}^{\mathrm{O}}y_t, {}^{\mathrm{O}}\theta_t)^\top$ The free space features are also defined on the local map and these features are compared with that of the global map. If corresponding features are found, candidate poses for global localization are determined.

Figure 3 illustrates a pose sampling scheme. Since the free-space feature has dominant orientation, these orientations are matched first. Then, a position on the map coordinates is determined. Candidate position
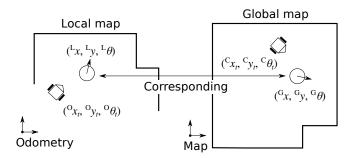
Figure 3: Pose sampling scheme from the global localizer.

and heading direction $^{\mathrm{C}}\mathbf{x}_t = (^{\mathrm{C}}x_t, ^{\mathrm{C}}y_t, ^{\mathrm{C}}\theta_t)^\top$ can be determined as follows

$$
\begin{aligned}
\Delta\theta_1 &= {}^{\mathrm{L}}\theta - {}^{\mathrm{G}}\theta, \quad \Delta\theta_2 = {}^{\mathrm{L}}\theta - {}^{\mathrm{O}}\theta_t \\
\Delta x &= {}^{\mathrm{L}}x - {}^{\mathrm{O}}x_t, \quad \Delta y = {}^{\mathrm{L}}y - {}^{\mathrm{O}}y_t \\
{}^{\mathrm{C}}x_t &= \Delta x \cos(\Delta\theta_1) - y \sin(\Delta\theta_1) + {}^{\mathrm{G}}x \\
{}^{\mathrm{C}}y_t &= \Delta x \sin(\Delta\theta_1) + y \cos(\Delta\theta_1) + {}^{\mathrm{G}}y \\
{}^{\mathrm{C}}\theta_t &= {}^{\mathrm{G}}\theta - \Delta\theta_2,
\end{aligned}
\tag{11}
$$

where G and L indicate the free-space features defined on the global and local maps, and x, y, and θ of them indicate the feature position and dominant orientation, respectively.

However, it is difficult to exactly determine the robot pose using Eq. (11) since the dominant orientation is roughly determined. Hence, poses for global localization are sampled by adding random noises to the candidate pose $^{\mathrm{C}}\mathbf{x}_t$. In addition, matching rate of the LiDAR measurements and map is computed using the sampled poses. If the matching rate is less than a threshold, the sampled pose is rejected. Furthermore, we found that the sampled poses sometimes face to opposite side to the ground truth pose. Hence, we also add opposite directional samples to the candidate poses.

4.5 Likelihood calculation

4.5.1 Class conditional measurement model

The likelihood of the particles used for estimating $p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, d_{1:t}, \mathbf{m})$, is calculated using two likelihood distributions denoted as $\sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{c}_t, \mathbf{m})p(\mathbf{c}_t)$ and $\sum_{s_t \in \mathcal{S}} p(d_t|\mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m})p(s_t)$. This sub-subsection describes how to calculate the first one. The second one is described in the next sub-subsection.

We first apply the independent assumption to the LiDAR measurements and decompose the likelihood distribution as shown in Eq. (12).

$$
\sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{c}_t, \mathbf{m})p(\mathbf{c}_t) = \prod_{k=1}^{K} \sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t^{[k]}|\mathbf{x}_t, c_t^{[k]}, \mathbf{m})p(c_t^{[k]}),
\tag{12}
$$

where K is the number of the LiDAR measurements. In this work, we consider two sensor measurement classes, i.e., $\mathcal{C} = \{\text{known, unknown}\}$, where known and unknown mean whether obstacles exist or do not exist on a given map. Namely, occupied cells are known obstacles and measurements obtained from other areas are unknown obstacles. We need to model two class conditional measurement models. It should be noted that the prior is uniformly set, i.e., $p(c_t^{[k]} = \text{known}) = p(c_t^{[k]} = \text{unknown}) = 0.5$, because there are no available information to estimate the prior.

When the known condition is given, the class conditional measurement model is implemented using the likelihood field model [Thrun et al.(2005)Thrun, Burgard, and Fox].

$$p(\mathbf{z}_t^{[k]}|\mathbf{x}_t, c_t^{[k]} = \text{known}, \mathbf{y}_t, \mathbf{m}) = p_{\text{LFM}}(\mathbf{z}_t^{[k]}|\mathbf{x}_t, \mathbf{m}) = \begin{pmatrix} z_{\text{hit}} \\ z_{\text{max}} \\ z_{\text{rand}} \end{pmatrix}^\top \cdot \begin{pmatrix} p_{\text{hit}}(\mathbf{z}_t^{[k]}|\mathbf{x}_t, \mathbf{m}) \\ p_{\text{max}}(\mathbf{z}_t^{[k]}|\mathbf{x}_t, \mathbf{m}) \\ p_{\text{rand}}(\mathbf{z}_t^{[k]}|\mathbf{x}_t, \mathbf{m}) \end{pmatrix}, \qquad (13)$$

where z_{hit}, z_{max}, and z_{rand} are arbitrary constants satisfying $z_{\text{hit}} + z_{\text{max}} + z_{\text{rand}} = 1$ and $p_{\text{hit}}(\cdot)$, $p_{\text{max}}(\cdot)$, and $p_{\text{rand}}(\cdot)$ are the measurement models related to the measurement of the known obstacles, maximum value, and random noise, respectively. The likelihood field model does not explicitly consider measuring dynamic obstacles ($p_{\text{max}}(\cdot)$ and $p_{\text{rand}}(\cdot)$ consider measuring noises). However, the use of the likelihood field model when the known condition is given is adequate because the given condition strongly restricts that the measurement is obtained from mapped obstacles.

When the unknown condition is given, the class conditional measurement model is implemented using the exponential distribution.

$$p(\mathbf{z}_t^{[k]}|\mathbf{x}_t, c_t^{[k]} = \text{unknown}, \mathbf{m}) = \frac{\lambda \exp(-\lambda r_t^{[k]})}{1 - \exp(-\lambda r_{\text{max}})}. \qquad (14)$$

where λ is the hyperparameter and r_{max} and $r_t^{[k]}$ are the maximum measurement and kth LiDAR measurement range. Predicting existence of unknown obstacles is difficult if we do not have any information. Hence, in the implementation, we assume that unknown obstacles are equally measured within the measurable range.

Eq. (12) is calculated as a sum of Eqs. (13) and (14). Eqs. (13) and (14) can be quickly calculated and Eq. (14) can be calculated without further information from that of used in the likelihood field model. Hence, calculation of the class conditional measurement model does not increase of computational and memory costs.

4.5.2 Decision model

Before describing the decision model, we first describe the localization state classifier. The classifier distinguishes whether localization has failed. To implement the classifier, mean absolute error (MAE) of residual errors is used in this study. It should be noted that the implementation is not limited with MAE. The residual errors are denoted as $\mathbf{e}_t = (e_t^{[1]}, e_t^{[2]}, ..., e_t^{[K]})^\top$, where $e_t^{[k]}$ is distance from kth LiDAR measurement point to the closest mapped obstacle. MAE is calculated using Eq. (15).

$$\text{MAE}_t = \frac{1}{\sum_{k=1}^{K} \mathbb{1}(e_t^{[k]} \leq e_{\text{max}})} \sum_{k=1}^{K} \mathbb{1}(e_t^{[k]} \leq e_{\text{max}}) e_t^{[k]}, \qquad (15)$$

where e_{max} is the maximum residual error and $\mathbb{1}(\cdot)$ is an indicator function which is equal to 1 when the condition within the bracket is true, and 0 otherwise. We set a threshold to MAE and classify as that localization has failed if MAE exceeds the threshold. In this implementation, MAE_t is equal to d_t in the graphical model shown in Fig. 1.

The decision model denoted as $p(d_t|\mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m})$ is needed to be modeled for two cases where localization has succeeded and failed. Here, we consider that success and failure localization results are positive and negative cases. In addition, the decision model is composed of true and false classification cases. Hence, following four distributions are used to calculate the decision model.

$$p_{\text{true}}(d_t \leq d_{\text{th}}|\mathbf{x}_t, s_t = \text{success}, \mathbf{z}_t, \mathbf{m}), \qquad (16)$$

$$p_{\text{false}}(d_t \leq d_{\text{th}}|\mathbf{x}_t, s_t = \text{failure}, \mathbf{z}_t, \mathbf{m}), \qquad (17)$$

$$p_{\text{true}}(d_t > d_{\text{th}}|\mathbf{x}_t, s_t = \text{failure}, \mathbf{z}_t, \mathbf{m}), \qquad (18)$$

$$p_{\text{false}}(d_t > d_{\text{th}}|\mathbf{x}_t, s_t = \text{success}, \mathbf{z}_t, \mathbf{m}), \qquad (19)$$

(a) $p(d_t|\mathbf{x}_t, s_t = \text{success}, \mathbf{z}_t, \mathbf{m})$ (b) $p(d_t|\mathbf{x}_t, s_t = \text{failure}, \mathbf{z}_t, \mathbf{m})$
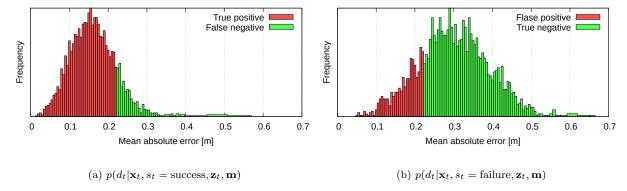
Figure 4: The decision models of the success (a) and failure (b) cases.
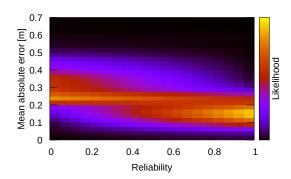

Figure 5: Likelihoods calculated using the decision model.

where Eqs. (16), (17), (18), and (19) are the probabilistic distributions over the true positive, false positive, true negative, and false negative cases, respectively. Note that $\int \left(p_{\text{true}}(d_t \leq d_{\text{th}}|\mathbf{x}_t, s_t = \text{success}, \mathbf{z}_t, \mathbf{m}) + p_{\text{false}}(d_t > d_{\text{th}}|\mathbf{x}_t, s_t = \text{success}, \mathbf{z}_t, \mathbf{m}) \right) \mathrm{d}d_t = 1$ and $\int \left(p_{\text{true}}(d_t > d_{\text{th}}|\mathbf{x}_t, s_t = \text{failure}, \mathbf{z}_t, \mathbf{m}) + p_{\text{false}}(d_t \leq d_{\text{th}}|\mathbf{x}_t, s_t = \text{failure}, \mathbf{z}_t, \mathbf{m}) \right) \mathrm{d}d_t = 1$.

Figure 4 shows the decision models modeled using histogram. In this case, we built the dataset using the 2D LiDAR simulation[2]. Because we used the simulation, the ground truth pose can be obtained. We aided noises to the ground truth pose and made success and failure poses. We set positional and angular thresholds to 0.02 m and 2 degrees and poses which exceeded either thresholds were classified as failure poses. The ground truth pose is not used as success poses because it does not have any error. The threshold to MAE, MAE_{TH}, was set to 0.225 m that performed most accurate classification on the dataset.

Likelihood calculation results by $\sum_{s_t \in \mathcal{S}} p(d_t|\mathbf{x}_t, s_t, \mathbf{z}_t, \mathbf{m})p(s_t)$ are shown in Fig. 5. As can be seen from the figure, likelihood is not to be 0 when MAE is large (or small) even if reliability is close to 1 (or 0). Consequently, uncertainty of MAE-based classification can be dealt with.

4.5.3 Predictive-distribution-based calculation

The likelihood of the particles sampled from global localization using the free-space feature is calculated using the predictive distribution denoted as $\int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m})\mathrm{d}\mathbf{x}_{t-1}$. In this

---

[2]https://github.com/NaokiAkai/AutoNavi

sub-subsection, we denote the predictive distribution as $p_{\text{pred}}(\mathbf{x}_t)$. In PF, the predictive distribution is approximated by the particles updated using the motion model.

$$p_{\text{pred}}(\mathbf{x}_t) \simeq \sum_{i=1}^{{}^{\text{P}}M} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}), \tag{20}$$

where ${}^{\text{P}}M$ is the number of the particles used for pose tracking. To realize stable localization, likelihood distribution should be smooth; however, the distribution shown in Eq. (20) is discrete. Hence, we assume that Eq. (20) can be approximated by a Gaussian mixture model as shown in Eq. (21).

$$\sum_{i=1}^{{}^{\text{P}}M} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}) \simeq \frac{1}{{}^{\text{P}}M} \sum_{i=1}^{{}^{\text{P}}M} \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_t^{[i]}, {}^{\text{P}}\Sigma\right), \tag{21}$$

where ${}^{\text{P}}\Sigma$ is an arbitrary covariance matrix.

However, likelihood of particles that sampled at far from the particles updated using the motion model, $\mathbf{x}_t^{[i]}$, is to be zero if the predictive distribution is modeled using Eq. (21). In other words, large localization error cannot be compensated. To overcome this problem, we model the predictive distribution as shown in Eq. (22).

$$p_{\text{pred}}(\mathbf{x}_t) \simeq \beta \frac{1}{{}^{\text{P}}M} \sum_{i=1}^{{}^{\text{P}}M} \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_t^{[i]}, {}^{\text{P}}\Sigma\right) + (1 - \beta)\text{unif}(\mathbf{m}), \tag{22}$$

where β is a positive constant including from 0 to 1 and $\text{unif}(\mathbf{m})$ is uniform distribution defined on areas where the robot can exist, that is, the free space. In the implementation, $\text{unif}(\mathbf{m})$ is approximated with a small constant value.

4.5.4   Likelihoods

In the presented localization method, there are two particle sets. The first set is used for pose tracking and the second one is sampled from global localization using the free-space feature. We denote each the particle set as ${}^{\text{P}}\mathbf{s}_t^{[i]} = ({}^{\text{P}}\mathbf{x}_t^{[i]}, {}^{\text{P}}\omega_t^{[i]})$ $(i = 1, 2, ..., {}^{\text{P}}M)$ and ${}^{\text{G}}\mathbf{s}_t^{[i]} = ({}^{\text{G}}\mathbf{x}_t^{[i]}, {}^{\text{G}}\omega_t^{[i]})$ $(i = 1, 2, ..., {}^{\text{G}}M)$. The likelihoods of these particles are calculated using Eqs. (23) and (24).

$$
{}^{\text{P}}\omega_t^{[i]} = \eta \sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t|{}^{\text{P}}\mathbf{x}_t^{[i]}, \mathbf{c}_t, \mathbf{m})p(\mathbf{c}_t) \sum_{{}^{\text{P}}s_t^{[i]} \in \mathcal{S}} p(d_t|{}^{\text{P}}\mathbf{x}_t^{[i]}, {}^{\text{P}}s_t^{[i]}, \mathbf{z}_t, \mathbf{m})p({}^{\text{P}}s_t^{[i]}), \tag{23}
$$

$$
{}^{\text{G}}\omega_t^{[i]} = \eta \frac{\sum_{\mathbf{c}_t \in \mathcal{C}} p(\mathbf{z}_t|{}^{\text{G}}\mathbf{x}_t^{[i]}, \mathbf{c}_t, \mathbf{m})p(\mathbf{c}_t) \sum_{{}^{\text{G}}s_t^{[i]} \in \mathcal{S}} p(d_t|{}^{\text{G}}\mathbf{x}_t^{[i]}, {}^{\text{G}}s_t^{[i]}, \mathbf{z}_t, \mathbf{m})p({}^{\text{G}}s_t^{[i]})}{\frac{1}{{}^{\text{G}}M} \sum_{j=1}^{{}^{\text{G}}M} \delta({}^{\text{G}}\mathbf{x}_t^{[i]} - {}^{\text{G}}\mathbf{x}_t^{[j]})} \cdot p_{\text{pred}}({}^{\text{G}}\mathbf{x}_t^{[i]}). \tag{24}
$$

It should be noted that the particles sampled from global localization do not have past information. Hence, the probability regarding the localization state is set to uniform, i.e., $p({}^{\text{G}}s_t^{[i]} = \text{success}) = p({}^{\text{G}}s_t^{[i]} = \text{failure}) = 0.5$.

4.6   Estimation

4.6.1   Robot pose

The robot pose is estimated as weight average of the two particle sets.

$$\mathbf{x}_t = \sum_{i=1}^{{}^{\text{P}}M} {}^{\text{P}}\omega_t^{[i]}{}^{\text{P}}\mathbf{x_t}^{[i]} + \sum_{i=1}^{{}^{\text{G}}M} {}^{\text{G}}\omega_t^{[i]}{}^{\text{G}}\mathbf{x_t}^{[i]}. \tag{25}$$

The likelihoods are normalized before computing Eq. (25), i.e., $\sum_{i=1}^{^\mathrm{P}M} {}^\mathrm{P}\omega_t^{[i]} + \sum_{i=1}^{^\mathrm{G}M} {}^\mathrm{G}\omega_t^{[i]} = 1$.

4.6.2 Sensor measurement classes

Through the likelihood calculation process, the particle with the maximum likelihood is extracted. We denote its pose as $^\mathrm{ML}\mathbf{x}_t$. The probabilistic distribution over the sensor measurement classes is calculated as shown in Eq. (26).

$$\eta p(\mathbf{z}_t | {}^\mathrm{ML}\mathbf{x}_t, {}^\mathrm{ML}\mathbf{c}_t, \mathbf{m}) p({}^\mathrm{ML}\mathbf{c}_t). \tag{26}$$

This can be calculated using Eqs. (13) and (14) and these can be quickly calculated. Hence, this calculation is performed after the likelihood calculation to reduce the memory cost. A measurement satisfying $p({}^\mathrm{ML}c_t^{[k]} = \mathrm{unknown}) > \chi$ is detected as an unknown obstacle, where χ is an arbitrary threshold including from 0 to 1.

4.6.3 Reliability

To estimate the reliability, we also use the maximum likelihood particle. The probabilistic distribution over the reliability is calculated as shown in Eq. (27).

$$\eta p(d_t | {}^\mathrm{ML}\mathbf{x}_t, {}^\mathrm{ML}s_t, \mathbf{z}_t, \mathbf{m}) p({}^\mathrm{ML}s_t). \tag{27}$$

$\eta p(d_t | {}^\mathrm{ML}\mathbf{x}_t, {}^\mathrm{ML}s_t = \mathrm{success}, \mathbf{z}_t, \mathbf{m}) p({}^\mathrm{ML}s_t = \mathrm{success})$ is regarded as the estimated reliability.

4.7 Re-sampling

To perform re-sampling, we first calculate values shown in Eq. (28).

$$b^{[i]} = \begin{cases} \sum_{j=1}^{i} {}^\mathrm{P}\omega_t^{[j]} & \text{if } i \leq {}^\mathrm{G}M, \\ \sum_{j=1}^{^\mathrm{G}M} {}^\mathrm{P}\omega_t^{[j]} + \sum_{j={}^\mathrm{G}M+1}^{i} {}^\mathrm{P}\omega_t^{[j-{}^\mathrm{G}M]} & \text{otherwise.} \end{cases} \tag{28}$$

Note that $0 \leq b^{[i]} \leq 1$ and $(i = 1, 2, ..., {}^\mathrm{G}M + {}^\mathrm{P}M)$. Then, random values including from 0 to 1, $r^{[i]}$, are generated. Re-sampling is performed based on Eq. (29).

$$\mathbf{x}_t^{[i]} \leftarrow \begin{cases} {}^\mathrm{P}\mathbf{x}_t^{[j]} & \text{if } r^{[i]} \leq b^{[j]} \text{ and } j \leq {}^\mathrm{G}M, \\ {}^\mathrm{G}\mathbf{x}_t^{[j-{}^\mathrm{G}M]} & \text{if } r^{[i]} \leq b^{[j]} \text{ and } j > {}^\mathrm{G}M. \end{cases} \tag{29}$$

In addition, likelihood is uniformly reset after re-sampling, i.e., ${}^\mathrm{P}\omega_t^{[i]} = {}^\mathrm{G}\omega_t^{[i]} = 1/({}^\mathrm{G}M + {}^\mathrm{P}M)$. Only ${}^\mathrm{G}M$ particles are re-sampled because the re-sampled particles are used for pose tracking.

5 Simulation experiments

This section describes experimental results using simulation. The simulation reffered at Section 4.5.2 was also used for the experiments.

5.1 Robustness to environment changes

We first tested robustness to environment changes. The main idea to improve robustness is to introduce the class conditional measurement model denoted as $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{c}_t, \mathbf{m})$ for likelihood calculation. Thus, we compared likelihood calculation results in dynamic situations. We refer the measurement model to CCMM in this section.

(a) Environment in which LFM was used

(b) Environment in which CCMM was used

(c) A likelihood map by LFM
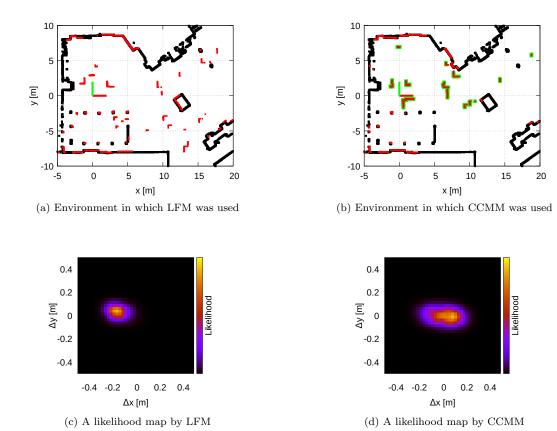
(d) A likelihood map by CCMM

Figure 6: Comparison of likelihood maps by LFM and CCMM in dynamic environments. The black, red, and green points of the top figures are the map, scan, and unknown scan points, respectively. The robot locates $(x, y) = (0, 0)$ and faces to x-axis parallelly in both the cases. The axes of the likelihood maps represent differences from the ground truth. Note that the heading direction is the same to that of the ground truth.

In MCL, the likelihood field model [Thrun et al.(2005)Thrun, Burgard, and Fox] is often used and we refer it to LFM. LFM is shown in Eq. (13). LFM is cost efficient and enables to calculate smooth likelihood distribution. However, LFM does not consider large environment changes. We compare likelihood calculation performance by LFM and CCMM.

Figure 6a and 6b show simulation environments in which LFM and CCMM were used, respectively. The black and red points indicate map and LiDAR measurement points. In this simulation, we simulate moving obstacles. The moving obstacles can be seen from the LiDAR points existing areas in which there are no map points. Because CCMM also estimates unknown measurements, that is, do not exist on the map, the unknown measurements are illustrated in Fig. 6b with the green. As can be seen from Fig. 6b, the moving obstacles were classified as the unknown obstacles.

Figure 6c and 6d show the likelihood maps around the ground truth. In both the cases, the ground truth is $(x, y, \theta) = (0, 0, 0)$. The axes of the likelihood field maps represent differences from the ground truth. Hence, we can say that likelihood can be robustly calculated if values close to $(\Delta x, \Delta y) = (0, 0)$ are high. Note that the heading angle is the same to that of the ground truth.

LFM could not generate accurate likelihood distribution; however, CCMM could generate accurate distribution. In addition, CCMM can be rapidly calculated. Average likelihood calculation time and its standard deviation were 12.63 msec and 3.95 msec when the number of the particles was set to 500. When LFM is used with the same condition, average and standard deviation are 9.36 msec and 3.20 msec. In both

the cases, calculation was processed with single thread. Furthermore, CCMM does not require additional information to calculate LFM, i.e., those memory costs are the same. From these results, we could confirm that CCMM can increase localization robustness to environment changes without increasing of memory and computational costs.

5.2 Reliability estimation

We tested reliability estimation performance. In the test, we adjusted the noise parameters shown in Eq. (9) to control localization performance. In other words, localization failures are forcibly occurred owing to misestimate of moving uncertainty. In addition, the moving obstacles were also simulated as shown in Fig. 6.

The left- and right-hand figures of Fig. 7 show the reliability estimation results in cases where localization has succeeded and failed during the test, i.e., the left- and right-hand figures are the results with the correct and incorrect noise parameters. In both the cases, the robot followed the same given path. Figures 7a and 7b indicate the ground truth (red) and estimated (blue) trajectories. In Fig. 7a, the estimate and the ground truth are the almost same, but these are different in Fig. 7b. Figures 7c and 7d show the positional (red) and angular (blue) errors of the estimate from the ground truth in Figs. 7a and 7b. Figures 7e and 7f show the estimated reliability (red) and MAE value of the maximum likelihood particle. The black line shown in Figs. 7e and 7f indicates the threshold set to MAE.

In the success case shown in the left-hand, the estimation errors were always small and the reliability was always close to 100 %. However, MAE sometimes exceeded the threshold around 70 sec. Nevertheless, the presented method recognized that localization successfully worked in such a case. This is because that the presented method estimates the reliability based on Bayes' theorem, i.e., the reliability can be smoothly estimated even when MAE suddenly exceeded the threshold. The reliability estimation can contribute to improve the classification robustness more than directly using the MAE-based classifier.

In the failure case shown in the right-hand, MAE exceeded the threshold many times. In particular around 40 sec, MAE completely exceeded the threshold and the presented method estimated that the estimate is unreliable, that is, 0 % reliability. Hence, the estimated reliability can be used for describing whether localization successfully works or not.

From 20 sec to 30 sec, the MAE values were less than and exceeded the threshold repeatedly. In such a case, the neutral reliability value was estimated. Such a neutral estimation cannot be obtained if the MAE-based classifier is directly used as the classifier. This is also an advantage to estimate the reliability based on Bayes' theorem. Based on such neutral estimation, we can consider variety strategy for safety guarantee.

5.3 Global localization and fusion

We first tested the pose sampling performance using the free-space feature. Figure 8 shows the pose sampling result. The blue and gray arrows indicate the ground truth and sampled poses. The black and red points indicate the map and LiDAR measurement points. The LiDAR measurements were plotted from the ground truth. As can be seen from the figure, the pose sampling cannot be accurately performed. However, some poses were generated around the ground truth. From the results, we can say that a sophisticated fusion manner is necessary to utilize the pose sampling results.

Then, we tested the failure recovery performance using the importance-sampling-based fusion. Figure 9 shows the failure recovery result. The blue, red, and gray arrows indicate the ground truth, estimate by the presented method, and the sampled poses. Figure 9a shows the localization failure result, and Fig. 9b shows the case after performing the fusion. As can be seen from the figure, we could confirm that the importance-sampling-based fusion enables to quickly recover from the localization failure.

(a) Estimated trajectory in the success case

(b) Estimated trajectory in the failure case

(c) Estimation errors in the success case

(d) Estimation errors in the failure case

(e) Reliability and MAE in the success case

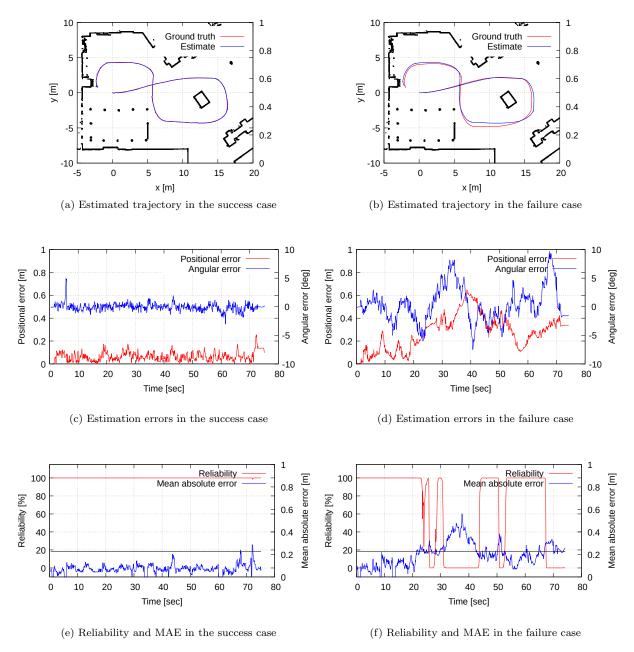(f) Reliability and MAE in the failure case

Figure 7: Reliability estimation results in the success (left) and failure (right) cases.

Finally, we tested the fusion performance with and without the importance sampling while the robot moving. In the cases, the robot also followed the same given path that was used in the tests shown in Fig. 7. Figure 10 shows the comparison results of the fusion performance. The blue, red, and green lines indicate the ground truth, MCL estimate, and the estimate using the free-space-feature-based global localization. It should be noted that the estimate of global localization is the average of the sampled poses. In this comparison, we also simulated dynamic obstacles as shown in Fig. 6.

In the result shown in Fig. 10a, the importance sampling was used. The pose sampling could not be accurately performed as shown in Fig. 8; however, the estimate by the presented method could smoothly follow the ground truth. In the result shown in Fig. 10b, the importance sampling was not used. If the
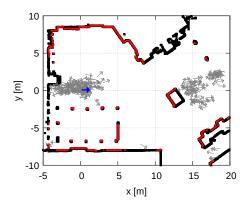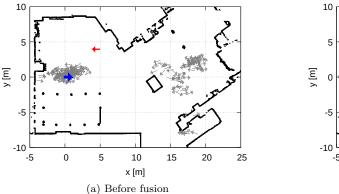
Figure 8: A pose sampling result based on the free-space feature. The black and red points indicate the map and LiDAR measurement points. The blue and gray arrows indicate the ground true and sampled poses.
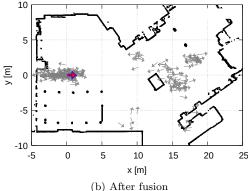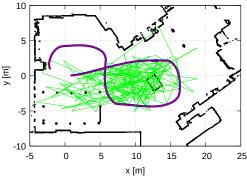

(a) Before fusion

(b) After fusion

Figure 9: A failure recovery result using the importance-sampling-based fusion. The blue, red, and gray arrows indicate the ground truth, estimate by the presented method, and the sampled poses. The left-hand figure shows the localization failure result, and the right-hand figure shows the case after performing the fusion.
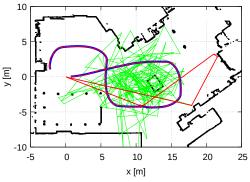
importance sampling was not used, the estimates sometimes jumped. In the likelihood calculation, the measurement-based likelihood calculation, i.e., the class conditional measurement model in the presented method, has the dominant role. Consequently, pose tracking can work even if the importance sampling is not used. However, the global-localization-based pose sampling generates poses which make the better match of the LiDAR measurements and map. In such a case, sudden jump cannot be prevented if the importance sampling is not used. However, the presented method could prevent such jumps. From the comparison, we could confirm that the use of the importance sampling contributes to mitigate influence of the noisy estimate by the free-space-feature-based pose sampling.

6 Experiments

In this section, we describe experimental results using public available dataset and our own platform. We used amcl[3] that is a popular localization package in Robot Operating System (ROS) as a comparison method in all the experiments conducted in this section. Videos of these experiments can be seen at `https://www.youtube.com/watch?v=wsoXvUgJvWk`.

---

[3] `http://wiki.ros.org/amcl`

(a) Estimated trajectory in the success case   (b) Estimated trajectory in the failure case

Figure 10: Pose fusion results with (left) and without (right) the importance sampling. The blue, red, and green lines indicate the ground truth, MCL estimate, and the average of the pose sampling results. The estimate of global localization is the average of the sampled poses.

6.1 Experiments using dataset

We used Pre-2014 Robotics 2D-Laser Datasets[4] for the dataset-based experiments. We selected two datasets; MIT CSAIL and Intel Research Lab. We first built environment maps using gmaping[5] and tested the localization methods with wrong initial pose. Note that we used this software[6] to use the datasets in ROS.

We used amcl as a comparison method. A function used in augmented MCL [Gutmann and Fox(2002)] is implemented in amcl. This function enables to detect localization failure by observing history of likelihoods. In addition, this function enables to perform re-localization by randomly sampling the particles on the free space if localization failure is detected. Hence, we compared pose tracking and re-localization performances of the presented method with that of amcl. However, amcl does not have an estimation function of localization correctness. We will show the reliability estimation performance by the presented method to reveal its effectiveness.

Figures 11 and 12 show the experimental results on the MIT CSAIL and Intel Research Lab datasets, respectively. The left and middle figures are the trajectories estimated by the presented method and amcl. We refer the presented method to "Ours" in the captions of this section for notational convenience. When the localization results are jumped, that is, re-localization was performed, the trajectories are depicted with the dashed lines. The figures also show the reference trajectories that are estimated by the presented method with the correct initial pose in the right-hand. We confirmed that the presented method successfully tracked the robot pose while making the reference trajectories.

In the MIT CSAIL case, the initial position was correct, but the initial heading angle was different about 120 degrees. The presented method could immediately compensated the heading angle; however, amcl could not compensated the initial error quickly. After 340 seconds when the presented method re-localized, amcl could re-localize. In addition, the presented method achieved accurate pose tracking even though re-localization process was parallelly performed in all the time of the experiment.

In the Intel Research Lab case, the initial pose was completely different. The correct initial pose was $(x, y, \theta) = (0, 0, 0)$, but the given initial pose was $(x, y, \theta) = (5, -19, \pi)$. The presented method also quickly re-localized the robot pose; however, amcl could not re-localize the robot pose during the experiment. In addition, the presented method achieved accurate pose tracking. From these results, we could confirm that

---

[4]https://www.ipb.uni-bonn.de/datasets/

[5]http://wiki.ros.org/gmapping
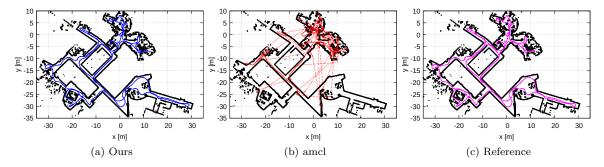
[6]https://github.com/NaokiAkai/carmen_player_ros

(a) Ours   (b) amcl   (c) Reference

Figure 11: The localization results on the MIT CSAIL dataset.
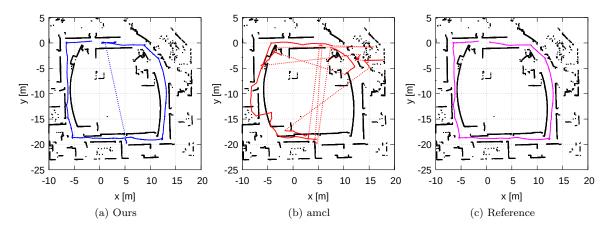

(a) Ours   (b) amcl   (c) Reference

Figure 12: The localization results on the Intel Research Lab dataset.

the presented method enables quick re-localization and accurate pose tracking simultaneously.

Furthermore, the reliability estimation results on the MIT CSAIL and Intel Research Lab datasets are shown in Fig. 13. In the MIT CSAIL case, low reliability was estimated when around the experiment starts. This estimation was correct since the wrong initial heading direction was given. High reliability was estimated after re-localization. The blue and black lines indicate MAE of the maximum likelihood particle and its threshold. As can be seen from the figure, MAE exceeded the threshold once; however, reliability was correctly estimated because of reliability was estimated using Bayesian filter.

In the Intel Research Lab case, low reliability was estimated sometimes. In that cases, the angular error was increased owing to inaccurate INS measurement[7]. However, the presented method immediately compensated the estimation error. From the cases, we can say that localization failures can be immediately detected based on the estimated reliability. In addition, high reliability was correctly estimated after the error compensation.

6.2   Experiments with our own platform

We also conducted experiments with our own experimental platform. In the experiments, we used Whill Model CR[8] equipped with UTM-30LX-EW[9]. The maximum measurement range of the LiDAR is 30 m and measurement angle and its resolution are 270 degrees and 0.25 degrees. We conducted experiments in an

---

[7]In the Intel Research Lab dataset, translational and angular velocities of the robot are not contained. We calculated the velocities from the pose information and these were sometimes inaccurate owing to significant small differences of timestamps.

[8]https://whill.inc/jp/model-cr

[9]https://www.generationrobots.com/en/401435-hokuyo-utm-30lx-ew-laser-range-finder-hokuyo.html
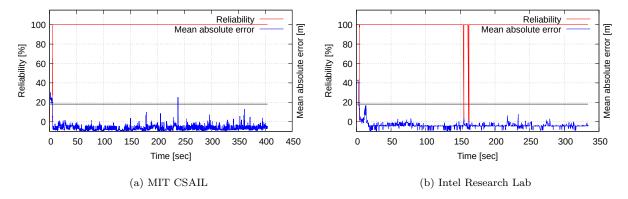
(a) MIT CSAIL

(b) Intel Research Lab

Figure 13: The reliability estimation results on the MIT CSAIL (a) and Intel Research Lab (b) datasets.

indoor environment with long corridors, i.e., there are areas where enough measurements cannot be obtained to correctly localize the robot pose. In general, long corridors are difficult areas for performing accurate localization when the maximum measurement range is short. In the experiments, we also compared the presented method with amcl.

Figure 14 shows two experimental results. In the results, the trajectories estimated by the presented method and amcl, and the reference trajectories are also shown. In the case shown in Figs. 14a and 14b, the robot started around $(x, y, \theta) = (-4, -35, \pi/2)$; however, the initial pose was set to $(x, y, \theta) = (0, 0, 0)$. The robot started from the bottom corridor, i.e., effective measurements to compensate the longitudinal error could not be observed. However, the presented method could re-localize quickly because the rooms shown in the bottom side were observed. After re-localization was successfully performed, the presented method accurately tracked the robot pose. However, amcl could not re-localize. In addition, amcl's estimate sometimes converged to wrong corrdidor side.

In the case shown in Figs. 14d and 14e, the robot started around $(x, y, \theta) = (0, 0, \pi)$; however, the initial pose was set to $(x, y, \theta) = (0, 0, 0)$. In this case, the presented method also quickly re-localized since the starting area is characteristic in this environment; however, amcl could not re-localize. From these results, we cloud also confrim that the presented method correctly works in areas with long corridors.

Finally, Fig. 15 shows the reliability estimation results in the above two cases. In both the cases, low reliability was estimated when the experiments start. These are correct results since the wrong initial pose was given in both the cases. After re-localization, high reliability was estimated. These results are also correct. We confirmed that reliability estimation by the presented method successfully works in the corridor environments.

6.3 Limitation

In this subsection, we discuss the limitations of the presented method regarding localization robustness, reliability estimation accuracy, and quickness of re-localization.

6.3.1 Localization robustness

To improve localization robustness to environment changes in the presented localization method, we used the class conditional measurement model. This model has the known and unknown classes as conditional variables. Owing to the conditional variables, we can separately model the measurements obtained from known and unknown obstacles. As a result, the measurements can be flexibly modeled and localization
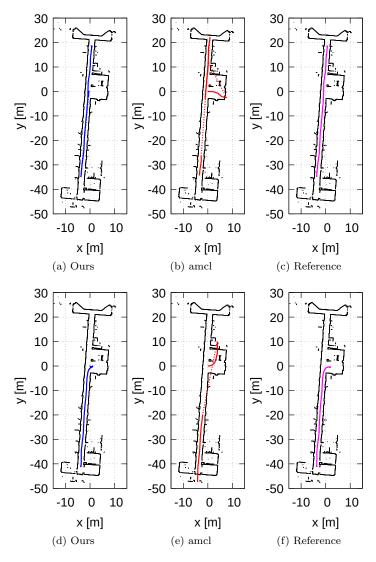
(a) Ours   (b) amcl   (c) Reference

(d) Ours   (e) amcl   (f) Reference

Figure 14: The localization results in the indoor with two long corridors.

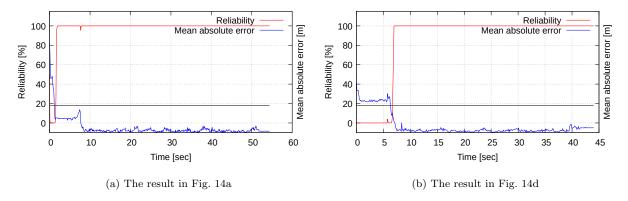(a) The result in Fig. 14a   (b) The result in Fig. 14d

Figure 15: The reliability estimation results in the top and bottom cases of Fig. 14.

robustness is improved without computational cost increasing. However, this model cannot contribute to realizing guarantee of localization performance since environment changes cannot be perfectly modeled even if the model is used. Hence, the presented method includes the reliability estimation function. We strongly note that reliable localization that we consider in this study cannot be realized with only robustness improvement.

The class conditional measurement model has similar effect to M-estimation [Hartley and Zisserman(2003)]. M-estimation uses a weight function to mitigate influence of outliers. This weight function gives correspondences having large distance a small weight and their effects are mitigated in the optimization process. Similar performance to this mitigation is realized owing to the use of the the unknown class in the conditional measurement model. M-estimation indeed improves robustness; however, it cannot work if initial estimate is far from the ground truth because almost all measurements are considered to be outliers and optimization does not work. In other words, there is a trade-off relationship between robustness and accuracy in the use of M-estimation. The class conditional measurement model also yields this negative effect. Hence, initial guess for localization have to be accurately estimated to utilize the class conditional measurement model.

6.3.2 Reliability estimation accuracy

In the presented localization method, reliability estimation is performed using the MAE-based classifier. This classification is performed based on whether the MAE value exceeds the threshold or not. Of course, this kind of simple-threshold-based classification does not stably work. However, we achieved stable reliability estimation because the reliability estimation process is performed based on the Bayesian filtering. Here, stability means ability to cope with noisy classification results.

However, it is hard for the presented method to improve classification accuracy of localization correctness. Interpretation of reliability estimated by the presented method is that *a predicted probability over what localization successfully works in online based on statistical performance of a used classifier*. Hence, the estimated reliability cannot tell us whether the localization results is close to the ground truth. The Bayesian filtering indeed contributes to improve stability of estimate because it considers the statistical performance of the classifier. However, reliability estimation accuracy depends on accuracy of a used classifier. An accurate classifier is necessary to improve reliability estimation accuracy and simple threshold-based classifier such as MAE-based one might be insufficient in complex scenes.

In [Akai et al.(2019)Akai, Morales, Hirayama, and Murase], we presented accurate localization correctness classifier. However, it is impossible to integrate the classifier with the graphical model shown in Fig. 1 because the computational cost of the classifier is high. To realize these integration, a new graphical model is required.

6.3.3 Quickness of re-localization

In this section, we conducted the localization experiments with wrong initial poses. The presented method achieved quick re-localization from the wrong poses. However, the re-localization performance depends on pose sampling based on used global localization. The global localization method used in the presented method utilizes features defined on the free space. In other words, the global localization performance is affected by environment changes since the features are strongly affected by shape of environment. We have to improve global localization performance for reliable quick re-localization.

The use of importance sampling improves robustness to miss sampling by global localization. However, evaluation using the predictive distribution has limitation. Eq. (22) contains the uniform distribution, $\text{unif}(\cdot)$. This uniform distribution has effect to achieve re-localization from miss estimation including large error. However, Eq. (22) approximates the predictive distribution, $\int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, d_{1:t-1}, \mathbf{m})\mathrm{d}\mathbf{x}_{t-1}$, and the use of the uniform distribution means that the robot could exist all the areas even if their movement is predicted using the motion model. Hence, there is a possibility where the robot jumps even though localization successfully works. This trade off relationship is limitation of the importance-sampling-based

re-localization. To remove this limitation, accurate global localization is necessary. However, in general, global localization does not stably work more than pose tracking. We need to improve the global localization performance to realize accurate and quick re-localization. Or, we need to estimate exact criterion to use global localization or not such as reliability estimation presented in our proposal.

7   Conclusion

This paper has presented reliable MCL that can (1) robustly work in dynamic environments, (2) immediately detect localization failure by estimating reliability, and (3) quickly re-localize if estimate has failed. To achieve these functions, we presented a novel graphical model and formulate the Bayesian filtering for the simultaneous localization, sensor measurement class estimation, and reliability estimation problem. We used Rao-Blackwellized particle filter to implement the simultaneous estimation system.

We conducted three types of experiments, that are, simulation-, dataset-, and our-own-platform-based experiments. Through the simulation experiments, we numerically showed that the presented method can achieve three things mentioned above. In addition, we showed that the presented method outperforms the traditional approaches. In the remaining two experiments, we compared the presented method with amcl that is a popular localization package in ROS. The comparison results showed that the presented method achives fast re-localization more than amcl. In addition, we confirmed that the presented method can accurately perform pose tracking and reliability estimation.

For the future work, we will try to propose a new graphical model that enables to integrate more accurate localization correctness classifier into the reliability estimation model in order to realize a practical reliability estimator.

Acknowledgments

References

N. Akai, L. Y. Morales, and H. Murase. Mobile robot localization considering class of sensor observations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3159–3166, 2018a.

N. Akai, L. Y. Morales, and H. Murase. Reliability estimation of vehicle localization result. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 740–747, 2018b.

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.

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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6469–6475, 2020.

N. Akai, Y. Akagi, T. Hirayama, T. Morikawa, and H. Murase. Detection of localization failures using Markov random fields with fully connected latent variables for safe lidar-based automated driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17130–17142, 2022.

J. Al Hage, P. Xu, P. Bonnifait, and J. Ibanez-Guzman. Localization integrity for intelligent vehicles through fault detection and position error characterization. *IEEE Transactions on Intelligent Transportation Systems*, 23(4):2978–2990, 2022. doi: 10.1109/TITS.2020.3027433.

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.

Z. Alsayed, G. Bresson, Verroust-Blondet A, and F. Nashashibi. Failure detection for laser-based SLAM in urban and peri-urban environments. In *2017 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 126–132, 2017.

Z. Alsayed, G. Bresson, Verroust-Blondet A, and F. Nashashibi. 2D SLAM correction prediction in large scale urban environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5167–5174, 2018.

D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *2002 Uncertainty in Artificial Intelligence (UAI)*, pages 10–17, 2002.

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.

P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *2005 Robotics: Science and Systems (RSS)*, pages 17–24, 2005.

P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *2003 IEEE/RSJ Intelligent Robots and Systems (IROS)*, pages 2743–2748, 2003.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006. ISBN 0387310738.

S. Brechtel, T. Gindele, and R. Dillmann. Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3932–3938, 2010.

W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1–2):3–55, 1999.

F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *1999 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1322–1328, 1999.

D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

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

J. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 454–459, 2002.

D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *2003 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1557–1563, 2003.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

L.T. Hsu. GNSS mulitpath detection using a machine learning approach. In *2017 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1414–1419, 2017.

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

A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

J. Kim and W. Chung. Robust localization of mobile robots considering reliability of lidar measurements. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6491–6496, 2018.

S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *2000 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1225–1232, 2000.

J. P. Mendoza, M. M. Veloso, and R. Simmons. Mobile robot fault detection based on redundant information statistics. *Carnegie Mellon University*, 2018.

D. Meyer-Delius, J. Hess, G. Grisetti, and W. Burgard. Temporary maps for robust localization in semi-static environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5750–5755, 2010.

D. Meyer-Delius, M. Beinhofer, and W. Burgard. Occupancy grid models for robot mapping in changing environments. In *2012 AAAI Conference on Artificial Intelligence (AAAI)*, pages 2024–2030, 2012.

A. Millane, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena. Free-space features: Global localization in 2D laser SLAM using distance function maps. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1271–1277, 2019. doi: 10.1109/IROS40897.2019.8967683.

M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *2002 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 695–701, 2002.

L. Montesano, J. Minguez, and L. Montano. Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. In *2005 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4556–4562, 2003.

S. Nobili, G. Tinchev, and M. Fallon. Predicting alignment risk to prevent localization failure. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1003–1010, 2018.

S. Olufs and M. Vincze. An efficient area-based observation model for monte-carlo robot localization. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13–20, 2009.

M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Integrity of map-matching algorithms. *Transportation Research Part C: Emerging Technologies*, 14(4):283–302, 2006.

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *2001 International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

J. Saarinen, H. Andreasson, and A. J. Lilienthal. Independent markov chain occupancy grid maps for representation of dynamic environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3489–3495, 2012.

D. Schulz, D. Fox, and J. Hightower. People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filters. In *2003 International Joint Conference on Artificial Intelligence (IJCAI)*, pages 921–926, 2003.

L. Silva, O. R. P. Bellon, and K. L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):762–776, 2005.

L. Sun, D. Adolfsson, M. Magnusson, H. Andreasson, I. Posner, and T. Duckett. Localising faster: Efficient and precise lidar-based robot localisation in large-scale environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4386–4392, 2020.

P. Sundvall and P. Jensfelt. Fault detection for mobile robots using redundant positioning systems. In *2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3781–3786, 2006.

E. Takeuchi, K. Ohno, and S. Tadokoro. Robust localization method based on free-space observation model using 3D-map. In *2010 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 973–979, 2010.

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

S. Thrun, J. Langford, and V. Verma. Risk sensitive particle filters. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 961–968, 2001b.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005. ISBN 0262201623.

G. D. Tipaldi, D. Meyer-Delius, and W. Burgard. Lifelong localization in changing environments. *International Journal of Robotics Research (IJRR)*, 32(14):1662–1678, 2013.

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 *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2481–2486, 2004.

R. Valencia, J. Saarinen, H. Andreasson, J. Vallvé, J. Andrade-Cetto, and A. J. Lilienthal. Localization in highly dynamic environments using dual-timescale NDT-MCL. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3956–3962, 2014.

C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *International Journal of Robotics Research (IJRR)*, 26(9):889–916, 2007.

D. F. Wolf and G. S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1):53–65, 2005.

S. W. Yang and C. C. Wang. Feasibility grids for localization and mapping in crowded urban scenes. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2322–2328, 2011.

W. Zhen, S. Zeng, and S. Scherer. Robust localization and localizability estimation with a rotating laser scanner. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6240–6245, 2017.