Reliable Monte Carlo localization for mobile robots

Naoki Akai

1Graduate Department of Aerospace Engineering, Graduate School of Engineering, Nagoya University, Nagoya, Japan
2LOCT Co., Ltd., Nagoya, Japan

Correspondence
Naoki Akai, Graduate School of Engineering, Nagoya University, Furo, Chikusa, Nagoya, Japan.
Email: akai@nagoya-u.jp

Funding information
Japan Science and Technology Agency

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 relocalization, 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 relocalization 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 two-dimensional LiDAR. Results show that reliable MCL that performs robust localization, self-failure detection, and quick failure recovery can be realized.

KEYWORDS

localization, mobile robots, modeling, probabilistic, reliability

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 two-dimensional (2D) LiDAR-based localization and present a method that enables to make localization more reliable. To realize reliable localization, we consider that localization must

1. robustly work in dynamic environments;
2. immediately detect failure of its estimate;
3. quickly relocalize the robot pose if failed, that is, 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 three-dimensional (3D)-LiDAR-based one.

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). Additionally, the sensor measurement class and reliability estimation methods we previously presented in Akai et al. (2018a,b) are integrated. The global localization process is implemented based on the free-space feature presented in Millane et al. (2019). Global localization is necessary to achieve relocalization 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) 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-, data set- 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 data set- 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 the website. The contribution is summarized as follows.

1. Presenting a new framework that integrates our previous proposals presented in Akai et al. (2018a, 2018b, 2020) and achieves reliable localization that simultaneously performs robust localization, reliability estimation, and quick relocalization.

2. 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. Sections 5 and 6 describe experimental results using simulation, data set and our own platform. Section 7 concludes this work.

2 | RELATED WORK

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

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). The traditional measurement models are the beam and likelihood field models presented in Thrun et al. (2005). 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.

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 (Anguelov et al., 2002; Burgard et al., 1999; Fox et al., 1999; Hähnel et al., 2003; Montemerlo et al., 2002; Schulz et al., 2003). 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 semidynamic 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 (2005) proposed an occupancy grid mapping method that separately maps static and dynamic parts. Montesano et al. (2003) 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; Meyer-Delius et al., 2012; Saarinen et al., 2012; Tipaldi et al., 2013). Wang et al. (2007) extended the probabilistic method and proposed a general framework for simultaneous localization, mapping, and dynamic object tracking. 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. (2010) and Valencia et al. (2014) 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 (2009) and Takeuchi et al. (2010) proposed the measurement model that utilizes the free space. Appropriate sensor measurement selection such as presented by 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). Our approach extends the graphical model and formulates the simultaneous localization and sensor measurement class estimation problem (see Figure 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 (2011). In their method, the

---

Footnote:
1https://github.com/NaokiAkai/als_ros

---

FIGURE 1 The graphical model for the pose tracking
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) 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, that is, robot pose. 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 & McKay, 1992) and normal distributions transform (NDT) (Biber & 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 & Levoy, 2001). However, insufficient performance of the threshold-based method when registration errors are small is presented in Silva et al. (2005). 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.

Qudus et al. (2006) 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 (2006) and Mendoza et al. (2018) 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. (2022) proposed a method for addressing localization integrity that combines measurement rejection and position error characterization. In this method, a multisensor data fusion with a fault detection and exclusion algorithm is constituted using a bank of information filters. Our approach does not use multisensor data for estimating reliability.

Modeling localization failure is not trivial. Hence, machine learning approaches have been applied recently. Almqvist et al. (2018) applied several threshold- and machine-learning-based methods to classify misaligned point clouds. Alsayed et al. (2017, 2018) presented a machine-learning-based failure detection method for 2D LiDAR SLAM. Similar approaches for global navigation satellite system (GNSS)-based localization is also presented in Hsu (2017). Zhen et al. (2017) 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, that is, it does not consider on-line situation. Nobili et al. (2018) 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 relocalization

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

Lensel and Veloso (2000) proposed the sensor resettings that enables to perform relocalization. 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 relocalization. Ueda et al. (2004) proposed the expansion resettings that also enables to perform relocalization. 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, Fox, et al. (2001) proposed mixture MCL (MMCL) that uses multiple proposal distributions and importance sampling to fuse particles generated from multiple distributions. In Thrun, Fox, et al. (2001), 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, Langford, et al. (2001) also proposed a risk sensitive particle filter (RSPF) that considers the risk estimated by Markov decision process. In Thrun, Langford, et al. (2001), 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. (2015) proposed PoseNet that directly estimates six-degree of freedom (DoF) pose from a camera image, that is, 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 (2015) showed that output of neural networks can be treated as posterior by approximating using variational inference. 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, that is, this sampling is similar to sampling from the measurement model. Sun et al. (2020) and we (Akai et al., 2020) 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 relocalization can be simultaneously achieved.

However, the use of E2E networks requires users preparation of training data set 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). This relocalization performance might not overcome the E2E-based one; however, we show that accurate and fast relocalization 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.

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 \( u \) and are used to predict the robot pose. Measurements of LiDAR are denoted as \( z \) and are used to perform matching with a map denoted as \( m \). In this work, we aim to estimate the current robot pose with sequences of \( u \) and \( z \), and \( m \). Additionally, we try to realize robust localization, reliability estimation, and quick relocalization simultaneously. Robust localization and reliability estimation are realized by the improved pose tracking algorithm described in the next section. Quick relocalization 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 \( x \), measurement classes \( c \), and localization state \( s \) are treated as the hidden variables. The INS and LiDAR measurements, \( u \) and \( z \), map \( m \), and output of a localization state classifier \( d \) are treated as the observable variables.

The LiDAR measurements are denoted as \( z = [z^{[1]}, z^{[2]}, ..., z^{[K]}] \), where \( z^{[k]} \) denotes \( k \)th measurement. The measurement classes \( c \) indicate category of the LiDAR measurements and denoted as \( c = [c^{[1]}, c^{[2]}, ..., c^{[K]}] \), where \( c^{[k]} \) corresponds to \( z^{[k]} \). In this work, we use two classes denoted as \( c^{[k]} \in C = \{ \text{known}, \text{unknown} \} \). 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). 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 \( s \in S = \{ \text{success}, \text{failure} \} \). 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, that is, \( p(s = \text{success}) \).

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 Equation (1).

\[
p(x_t, c_t, s_t | u_{1:t}, z_{1:t}, d_{1:t}, m),
\]

where \( 1 : t \) indicates time sequence, that is, \( u_{1:t} = (u_1, u_2, ..., u_t) \). Equation (1) can be decomposed using the multiplication theorem
\[
p(x_t, c_t, s_t|u_{t-1}, z_{t-1}, d_{t-1}, m) = p(x_t|u_{t-1}, z_{t-1}, d_{t-1}, m) \tag{2}
\]
\[
p(c_t|x_t, u_{t-1}, z_{t-1}, d_{t-1}, m) \tag{3}
\]
\[
p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m), \tag{4}
\]

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

Equations (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 Equation (2).

\[
\begin{align*}
p(x_t|u_{t-1}, z_{t-1}, d_{t-1}, m) & = p(x_t|u_{t-1}, z_{t-1}, d_{t-1}, m) p(d_t|x_t, u_{t-1}, z_{t-1}, d_{t-1}, m) \\ &= p(x_t|u_{t-1}, z_{t-1}, d_{t-1}, m) p(d_t|u_{t-1}, z_{t-1}, d_{t-1}, m) \\
&= \eta p(z_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m) p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m)
\end{align*}
\]

To change from Lines 2 to 3, D-separation (Bishop, 2006) was applied to the distributions over \( z_t \) and \( d_t \) to remove nonrelated conditional variables. The law of total probability is then applied to each distribution.

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

\[
\begin{align*}
p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m) & = \eta p(d_t|x_t, c_t, s_t, u_{t-1}, z_{t-1}, d_{t-1}, m) p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m) \\
&= \eta p(d_t|x_t, c_t, s_t, z_{t-1}, d_{t-1}, m) p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m)
\end{align*}
\]

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

\[
\begin{align*}
p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m) & = \sum_{s_t \in S} p(s_t|x_t, c_t, s_t, u_{t-1}, z_{t-1}, d_{t-1}, m) p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m) \\
&= \sum_{s_t \in S} p(s_t|x_t, u_{t-1}, z_{t-1}, d_{t-1}, m) p(s_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m)
\end{align*}
\]

To change from Lines 2 to 3, D-separation was applied to the distribution over \( s_t \) and \( x_t, c_t, u_{t-1}, \) and \( z_{t-1} \) was omitted from the distribution over \( s_t \) 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 Equation (5).

\[
\begin{align*}
p(z_t|x_t, c_t, m) & = \sum_{s_t \in S} p(z_t|x_t, c_t, m) p(c_t) \sum_{s_t \in S} p(d_t|x_t, s_t, z_{t-1}, m) p(s_t) \int p(x_t|x_t-1, u_t) p(x_t-1|u_{t-1}, z_{t-1}, d_{t-1-1}, m) dx_{t-1} \\
p(d_t|x_t, s_t, z_{t-1}, m) & = \sum_{s_t \in S} p(d_t|x_t, s_t, z_{t-1}, m) p(s_t) \int p(x_t|x_t-1, u_t) p(x_t-1|u_{t-1}, z_{t-1}, d_{t-1-1}, m) dx_{t-1}
\end{align*}
\]

In Equation (5), we have four important models denoted as \( p(z_t|x_t, c_t, m), p(d_t|x_t, s_t, z_{t-1}, m), p(x_t|x_t-1, u_t), \) and \( p(s_t|x_t, u_{t-1}), \) these models are referred to as the class conditional measurement, decision, motion, and reliability transition models, respectively. These models are detailed in Sections 4.2, 4.3, 4.5.1, and 4.5.2, respectively.

In this work, we use Rao–Blackwellized particle filter (RBPF) to estimate Equation (5). More precisely, Equation (2) is estimated using particle filter (PF) and Equations (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 Equation (6).

\[
p(x_{t-1}N+1, z_{t-1}, m), \tag{6}
\]

To change from Lines 2 to 3, D-separation was applied to the distribution over \( x_t, c_t, u_{t-1}, \) and \( z_{t-1} \) was omitted from the distribution over \( x_t, c_t, u_{t-1}, \) since the future conditions do not affect to the previous pose. The recursive update equation regarding the pose distribution is obtained.

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

\[
\begin{align*}
p(c_t|x_t, u_{t-1}, z_{t-1}, d_{t-1}, m) & = \eta p(z_t|x_t, c_t, u_{t-1}, z_{t-1}, d_{t-1}, m) p(c_t|x_t, u_{t-1}, z_{t-1}, d_{t-1}, m) \\
&= \eta p(z_t|x_t, c_t, m) p(c_t)
\end{align*}
\]
where \( N \) is the number of previous steps used for global localization. It should be noted that the INS measurement \( u \) is ignored if \( N = 0 \).

As shown in Equations (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 (Akai et al., 2022, 2019; Thrun et al., 2005). 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 Equation (6). Let \( q_M \) be a number of sampled poses and \( \omega(i) \) be a i-th sampled pose, we assume that the sampled poses approximate Equation (6) like the approximation by PF.

\[
p(x|u_{t-N-1:t}, z_{t-N:t}, m) \approx \frac{1}{q_M} \sum_{i=1}^{q_M} \delta(x_t - \omega(i)),
\]

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 Equation (7), we can use the distribution shown in Equation (6) as a proposal distribution, that is, 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 Equation (2). Hence, the likelihood of the particles sampled from Equation (6) is denoted as

\[
c_{\omega} = \frac{p(x_t|u_{t-1:t}, z_{t-1:t}, d_{t-1:t}, m)}{p(x_t|u_{t-N-1:t}, z_{t-N:t}, m)}
\]

\[
\approx \int \frac{1}{q_M} \sum_{i=1}^{q_M} \delta(x_t - \omega(i)) \cdot \int p(x_t|u_{t-1}, u_t)p(x_{t-1}|u_{t-1}, z_{t-1:t-1}, d_{t-1:t-1}, m)dx_{t-1}.
\]

where \( \int p(x_t|u_{t-1}, u_t)p(x_{t-1}|u_{t-1}, z_{t-1:t-1}, d_{t-1:t-1}, m)dx_{t-1} \) is referred to predictive distribution.

In the robot pose distribution shown in Equation (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 Equation (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 Equation (6). In this work, we use localization using the free-space feature presented in Millane et al. (2019). 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 Figure 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. resampling the particles,
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 \( s_t \) contains 2D pose, \( x_t, y_t \), and \( \theta_t \), and likelihood, \( \omega_t \). 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 Equation (2) is constant and is denoted as \( q_M \).

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 Equation (6). The number of these particles is not constant and is denoted as \( q_M \).

FIGURE 2 The localization system diagram
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 \( x_{t+1} = f(x_t, u_t) \), 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 \( u_t \). Update of the particles using the motion model is denoted as

\[
\hat{x}_t^{[i]} = f\left(x_{t-1}^{[i]}, u_t^{[i]}\right), \quad u_t^{[i]} \sim N(u_t, \Sigma(u_t)),
\]

where \( \Sigma \) 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, that is, \( \hat{r}_t \leq r_{t-1} \), 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 Equation (10).

\[
\hat{r}_t = r_{t-1} \left( 1 - \sum_l \alpha_l \Delta t^l \right),
\]

where \( \alpha_l \) is a positive arbitrary constant and \( \Delta t^l \) is displacement between the time steps measured by INS, for example, translational and angular displacements, \( \Delta d \) and \( \Delta \theta \), are used if the differential driving model is used.

In the implementation, we set all \( \alpha_l \) 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, for example, more than 10Hz. 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). In this section, we briefly describe the free-space feature. For more details, please see the literature.

#### 4.4.1 Free-space feature

Let \( m \) be an occupancy grid map (OGM) and is denoted as \( m = (m^1, m^2, \ldots, m^M) \), where \( m^i \in \{\text{occupied, free space, unknown}\} \) is a state of \( i \)-th 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 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, that is, 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 \( m \) in advance. In the localization phase, the LiDAR measurements are accumulated based on the INS measurements and a local map \( \hat{m} \) is built. In other words, the local map is built on the odometry frame. The pose of the odometry frame is denoted as \( \hat{O}_x = [\hat{x}_r, \hat{y}_r, \hat{\theta}_r]^T \) 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 and heading direction \( Cx_i = [x_i, y_i, \theta_i]^T \) can be determined as follows:

\[
\Delta \theta_1 = \theta - \theta_1, \Delta x = x - x_1, \Delta y = y - y_1,
\]

\[
C_x_i = \Delta x \cos(\Delta \theta) - \Delta y \sin(\Delta \theta) + C_x
\]

\[
C_y_i = \Delta x \sin(\Delta \theta) + \Delta y \cos(\Delta \theta) + C_y
\]

\[
C_{\theta_i} = \theta - \Delta \theta_2,
\]

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

However, it is difficult to exactly determine the robot pose using Equation (11) since the dominant orientation is roughly determined. Hence, poses for global localization are sampled by adding random noises to the candidate pose \( Cx_i \). 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 \( p(x|c_{1:t}, z_{1:t}, d_{1:t}, m) \), is calculated using two likelihood distributions denoted as \( \sum_{c_i \in \mathcal{C}} p(z_i|c_i, m)p(c_i) \) and \( \sum_{k=1}^{K} p(z_k|\{x_i, c_{i:k}\}, m)p(c_{i:k}) \). This section describes how to calculate the first one. The second one is described in the next section.

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

\[
\sum_{c_i \in \mathcal{C}} p(z_i|c_i, m)p(c_i) = \prod_{k=1}^{K} \sum_{c_{i:k} \in \mathcal{C}} p(z_k|c_{i:k}, m)p(c_{i:k}).
\]

where \( K \) is the number of the LiDAR measurements. In this work, we consider two sensor measurement classes, that is, \( \mathcal{C} = \{ \text{known}, \text{unknown} \} \), 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, that is, \( p(c_{i:k} = \text{known}) = p(c_{i:k} = \text{unknown}) = 0.5 \), 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).

\[
p(z_k|c_{i:k} = \text{known}, y_i, m) = p_{\text{LFM}}(z_k|c_{i:k}, x_i, m)
\]

\[
= \begin{bmatrix} z_{hit}^T \\
z_{max} \\
z_{rand} \\
\end{bmatrix},
\]

where \( z_{hit}, z_{max}, \) and \( z_{rand} \) are arbitrary constants satisfying \( z_{hit} + z_{max} + z_{rand} = 1 \) 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.

\[
p(z_k|c_{i:k} = \text{unknown}, m) = \lambda \exp(-\lambda r_{i:k})(1 - \exp(-\lambda r_{i:k})).
\]

where \( \lambda \) is the hyperparameter and \( r_{max} \) and \( r_{i:k} \) are the maximum measurement and \( k \)th 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.

Equation (12) is calculated as a sum of Equations (13) and (14). Equations (13) and (14) can be quickly calculated and Equation (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 \( e_i = \{e_i^{[1]}, e_i^{[2]}, ..., e_i^{[K]} \} \), where \( e_i^{[k]} \) is distance from \( k \) th LiDAR measurement point to the closest mapped obstacle. MAE is calculated using Equation (15).

\[
\text{MAE}_i = \frac{1}{\sum_{k=1}^{K} I(e_i^{[k]} \leq e_{max})} \sum_{k=1}^{K} I(e_i^{[k]} \leq e_{max}) e_i^{[k]},
\]

where \( e_{max} \) is the maximum residual error and \( I(\cdot) \) 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, is equal to $d_t$ in the graphical model shown in Figure 1. The decision model denoted as $p(d_i|x_t, s_t, z_t, m)$ 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.

\[
p_{\text{true}}(d_t \leq d_{th}, s_t = \text{success}, z_t, m), \tag{16}
\]

\[
p_{\text{false}}(d_t \leq d_{th}, s_t = \text{failure}, z_t, m), \tag{17}
\]

\[
p_{\text{true}}(d_t > d_{th}, s_t = \text{success}, z_t, m), \tag{18}
\]

\[
p_{\text{false}}(d_t > d_{th}, s_t = \text{failure}, z_t, m). \tag{19}
\]

where Equations (16), (17), (18), and (19) are the probabilistic distributions over the true positive, false positive, true negative, and false negative cases, respectively. Note that $\int p_{\text{true}}(d_t \leq d_{th}, s_t = \text{success}, z_t, m) + p_{\text{false}}(d_t > d_{th}, s_t = \text{failure}, z_t, m) \, dd_t = 1$ and $\int p_{\text{true}}(d_t > d_{th}, s_t = \text{success}, z_t, m) + p_{\text{false}}(d_t \leq d_{th}, s_t = \text{failure}, z_t, m) \, dd_t = 1$.

Figure 4 shows the decision models modeled using histogram. In this case, we built the data set using the 2D LiDAR simulation.\textsuperscript{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, $\text{MAE}_{\text{TH}}$, was set to 0.225 m that performed most accurate classification on the data set.

Likelihood calculation results by $\sum_{s_t \in S} p(d_i|x_t, s_t, z_t, m)p(s_t)$ are shown in Figure 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.

\textsuperscript{2}https://github.com/NaokiAkai/AutoNavi

FIGURE 4 The decision models of the success (a) and failure (b) cases

FIGURE 5 Likelihoods calculated using the decision model

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 $\int p(x|u|u_{t-1}, x_{t-1})p(x_{t-1}|u_{t-1}, z_{t-1}, d_{t-1}, m) \, dx_{t-1}$. In this section, we denote the predictive distribution as $p_{\text{pred}}(x_i)$. In PF, the predictive distribution is approximated by the particles updated using the motion model.

\[
p_{\text{pred}}(x_i) \approx \sum_{i=1}^{P_M} \delta(x_i - x_i^{[l]}). \tag{20}
\]

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

\[
\sum_{i=1}^{P_M} \delta(x_i - x_i^{[l]}) \approx \frac{1}{P_M} \sum_{i=1}^{P_M} N(x_i; x_i^{[l]}, \Sigma). \tag{21}
\]

where $\Sigma$ is an arbitrary covariance matrix.

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

\[ p_{\text{pred}}(x_i) = \beta \frac{1}{M} \sum_{i=1}^{M} \mathcal{N}(x_i; x_i^{(i)}, P_x) + (1 - \beta) \text{unif}(m), \]  

where \( \beta \) is a positive constant including from 0 to 1 and \( \text{unif}(m) \) is uniform distribution defined on areas where the robot can exist, that is, the free space. In the implementation, \( \text{unif}(m) \) 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 \( p_i^{(i)} = (x_i, \omega_i) \) and \( G_i^{(i)} = (x_i, \omega_i) \) for \( i = 1, 2, \ldots, M \). The likelihoods of these particles are calculated using Equations (23) and (24).

\[ p_{\omega_i} = \eta \sum_{c \in C} p(z_i, x_i^{(i)}, c, m) p(c) \sum_{p_i^{(i)}} p(d_i^{(i)}, z_i, m) p(s_i^{(i)}), \]  

\[ e_{\omega} = \frac{M}{\sum_{i=1}^{M} \sum_{p_i^{(i)}} p(d_i^{(i)}, x_i^{(i)}, z_m, m) p(s_i^{(i)})} \]  

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, that is, \( p(G_i^{(i)} = \text{success}) = p(G_i^{(i)} = \text{failure}) = 0.5 \).

4.6 | Estimation

4.6.1 | Robot pose

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

\[ x_i = \sum_{i=1}^{P_M} p_i^{(i)} x_i^{(i)} + \sum_{i=1}^{G_M} c_i^{(i)} x_i^{(i)}. \]  

The likelihoods are normalized before computing Equation (25), that is, \( \sum_{i=1}^{P_M} \omega_i^{(i)} + \sum_{i=1}^{G_M} \omega_i^{(i)} = 1 \).

4.6.2 | Sensor measurement classes

Through the likelihood calculation process, the particle with the maximum likelihood is extracted. We denote its pose as \( M_s x_i \).

The probabilistic distribution over the sensor measurement classes is calculated as shown in Equation (26).

\[ \eta p(z_i^{(i)} x_i^{(i)} M_s c_i^{(i)} m p(M_s c_i^{(i)}). \]  

This can be calculated using Equations (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 \( p(M_s c_i^{(i)} = \text{unknown}) > \chi \) is detected as an unknown obstacle, where \( \chi \) 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 Equation (27).

\[ \eta p(d_i^{(i)} x_i^{(i)} M_s z_i^{(i)} m p(M_s z_i^{(i)}). \]  

\[ \eta p(d_i^{(i)} x_i^{(i)} M_s z_i^{(i)} = \text{success}, z_i^{(i)} m p(M_s z_i^{(i)} = \text{success}) \]  


4.7 | Resampling

To perform resampling, we first calculate values shown in Equation (28).

\[ b^{(i)} = \begin{cases} \frac{p_i^{(i)}}{M} & \text{if } i \leq M, \\ \frac{1}{M} & \text{otherwise}. \end{cases} \]  

Note that \( 0 \leq b^{(i)} \leq 1 \) and \( i = 1, 2, \ldots, M + b^{(i)} \). Then, random values including from 0 to 1, \( r^{(i)} \), are generated. Resampling is performed based on Equation (29).

\[ x_i^{(i)} \leftarrow \begin{cases} x_i^{(i)} & \text{if } r^{(i)} \leq b^{(i)} \text{ and } j \leq M, \\ x_i^{(i)} c_i^{(i)} & \text{if } r^{(i)} \leq b^{(i)} \text{ and } j > M. \end{cases} \]  

In addition, likelihood is uniformly reset after resampling, that is, \( p_\omega^{(i)} = c_\omega^{(i)} = 1/(M + b^{(i)}) \). Only \( M \) particles are resampled because the resampled particles are used for pose tracking.

5 | SIMULATION EXPERIMENTS

This section describes experimental results using simulation. The simulation referred 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 \( p(z|x, c, m) \) for likelihood calculation. Thus, we compared likelihood calculation results in dynamic situations. We refer the measurement model to CCMM in this section.

In MCL, the likelihood field model (Thrun et al., 2005) is often used and we refer it to likelihood field model (LFM). LFM is shown in Equation (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 class conditional measurement model (CCMM).

Figure 6a,b 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 Figure 6b with the green. As can be seen from Figure 6b, the moving obstacles were classified as the unknown obstacles.

Figure 6c,d 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 and 3.95 ms 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 and 3.20 ms. In both the cases, calculation was processed with single thread. Furthermore, CCMM does not require additional information to calculate LFM, that is, 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.

**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.

(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. CCMM, LFM.
5.2 Reliability estimation

We tested reliability estimation performance. In the test, we adjusted the noise parameters shown in Equation (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 Figure 6.

The left- and right-hand figures of Figure 7 show the reliability estimation results in cases where localization has succeeded and failed during the test, that is, 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. Figure 7a,b indicate the ground truth (red) and estimated (blue) trajectories. In Figure 7a, the estimate and the ground truth are the almost same, but these are different in Figure 7b. Figure 7c,d show the positional (red) and angular (blue) errors of the estimate from the ground truth in Figure 7a,b. Figure 7e,f show the estimated reliability (red) and MAE value of the maximum likelihood particle. The black line shown in Figure 7e,f 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 70s. 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, that is, 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 40s, MAE completely exceeded the threshold and the presented method estimated that the estimate is unreliable, that is, 0% reliability. Hence, the estimated

![Figure 7](image-url)
reliability can be used for describing whether localization successfully works or not.

From 20 to 30 s, 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.

![Figure 8](image1.png)

**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.

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 Figure 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.

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 Figure 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 Figure 6.

In the result shown in Figure 10a, the importance sampling was used. The pose sampling could not be accurately performed as shown in Figure 8; however, the estimate by the presented method could smoothly follow the ground truth. In the result shown in Figure 10b, the importance sampling was not used. If the importance sampling was not used, the estimates sometimes jumped. In the likelihood calculation, the measurement-based likelihood calculation, that is, 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

![Figure 9](image2.png)

**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. (a) Before fusion and (b) after fusion.
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 data set and our own platform. We used amcl 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.

6.1 | Experiments using data set

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

We used amcl as a comparison method. A function used in augmented MCL (Gutmann & 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 relocalization by randomly sampling the particles on the free space if localization failure is detected. Hence, we compared pose tracking and relocalization 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 data sets, 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, relocalization 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°. The presented method could immediately compensated the heading angle; however, amcl could not compensated the initial error quickly. After 340s when the presented method relocalized, amcl could relocalize. In addition, the presented method achieved accurate pose tracking even though relocalization 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 $\mathbf{x}_0 = (0, 0, 0)$, but the given initial pose was $\mathbf{x}_0 = (5, -19, \pi)$. The presented method also quickly relocalized the robot pose; however, amcl could not relocalize the robot pose during the experiment. In addition, the presented method achieved accurate pose tracking. From these results, we could confirm that the presented method enables quick relocalization and accurate pose tracking simultaneously.

Furthermore, the reliability estimation results on the MIT CSAIL and Intel Research Lab data sets are shown in Figure 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 relocalization. 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.\textsuperscript{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.

\textbf{6.2} | Experiments with our own platform

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

\textsuperscript{7}In the Intel Research Lab data set, 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.

\textsuperscript{8}https://whill.inc/jp/model-cr

\textsuperscript{9}https://www.generationrobots.com/en/401435-hokuyo-utm-30lx-ew-laser-range-finder-hokuyo.html
and 0.25°. We conducted experiments in an indoor environment with long corridors, that is, 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 Figure 14a,b, 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, that is, effective measurements to compensate the longitudinal error could not be observed. However, the presented method could relocalize quickly because the rooms shown in the bottom side were observed. After relocalization was successfully performed, the presented method accurately tracked the robot pose. However, amcl could not relocalize. In addition, amcl's estimate sometimes converged to wrong corridor side.

In the case shown in Figure 14d,e, 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 relocalized since the starting area is characteristic in this environment; however, amcl could not relocalize. From these results, we could also confirm that the presented method correctly works in areas with long corridors.

Finally, Figure 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 relocalization, 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 section, we discuss the limitations of the presented method regarding localization robustness, reliability estimation accuracy, and quickness of relocalization.

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 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 & 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), we presented accurate localization correctness classifier. However, it is impossible to integrate the classifier with the graphical model shown in Figure 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 relocalization

In this section, we conducted the localization experiments with wrong initial poses. The presented method achieved quick relocalization from the wrong poses. However, the relocalization 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 relocalization.

The use of importance sampling improves robustness to miss sampling by global localization. However, evaluation using the predictive distribution has limitation. Equation (22) contains the uniform distribution, unif(·). This uniform distribution has effect to achieve relocalization from miss estimation including large error. However, Equation (22) approximates the predictive distribution, \( p(x|\tilde{x}_t, t|\tilde{u}_t, \tilde{d}_t, \tilde{m}) \), 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 relocalization. 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 relocalization. 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 relocalize 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 RBPF to implement the simultaneous estimation system.

We conducted three types of experiments, that are, simulation-, data set-, 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 achieves fast relocalization 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 to realize a practical reliability estimator.

ACKNOWLEDGMENT

This research was supported by the Japan Science and Technology Agency and KAKENHI 18K13727.

DATA AVAILABILITY STATEMENT

The data that supports the findings of this study are available in the supplementary material of this article.

ORCID

Naoki Akai  http://orcid.org/0000-0002-4883-080X

REFERENCES

Akai, N., Akagi, Y., Hirayama, T., Morikawa, T. & Murase, H. (2022) 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.

Akai, N., Hirayama, T. & Murase, H. (2020) 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). pp. 6469–6475.

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

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

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

AllHage, J., Xu, P., Bonnifait, P. & Ibanez-Guzman, J. (2022) Localization integrity for intelligent vehicles through fault detection and position error characterization. IEEE Transactions on Intelligent Transportation Systems, 23(4), 2978–2990.

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

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

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

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

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

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

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

Bishop, C.M. (2006) Pattern recognition and machine learning. Springer-Verlag.

Brechtel, S., Gindele, T. & Dillmann, R. (2010) Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments. In: 2010 IEEE International Conference on Robotics and Automation (ICRA). pp. 3932–3938.
Burgard, W., Cremers, A., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., Steiner, W. & Thrun, S. (1999) Experiences with an interactive museum tour-guide robot. Artificial Intelligence, 114:1–2, 3–55.

Dellaert, F., Fox, D., Burgard, W. & Thrun, S. (1999) Monte Carlo localization for mobile robots. In: 1999 IEEE International Conference on Robotics and Automation (ICRA), vol 2, pp. 1322–1328.

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

Gal, Y. & Ghahramani, Z. (2015) Bayesian convolutional neural networks with Bernoulli approximate variational inference. arXiv:1506.02158.

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

Hänkel, D., Triebel, R., Burgard, W. & Thrun, S. (2003) Map building with mobile robots in dynamic environments. In: 2003 IEEE International Conference on Robotics and Automation (ICRA), vol 2, pp. 1557–1563.

Hartley, R. & Zisserman, A. (2003) Multiple view geometry in computer vision. 2nd edition, New York, NY: Cambridge University Press.

Hsu, L. (2017) GNSS multipath detection using a machine learning approach. In: 2017 IEEE International Conference on Intelligent Transportation Systems (ITSC), pp. 1414–1419.

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

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

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

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

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

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

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

Millane, A., Olenyikova, H., Nieto, J., Siegwart, R. & Cadena, C. (2019) 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), pp. 1271–1277.

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

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

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

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

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

Rusinkiewicz, S. & Levoy, M. (2001) Efficient variants of the ICP algorithm. In: 2001 International Conference on 3-D Digital Imaging and Modeling. pp. 145–152.

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

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

Sila, L., Bellon, O.R.P. & Boyer, K.L. (2005) 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.

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

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

Takeuchi, E., Ohno, K. & Tedokoro, S. (2010) Robust localization method based on free-space observation model using 3D-map. In: 2010 IEEE International Conference on Robotics and Biomimetics (ROBBIO), pp. 973–979.

Thrun, S., Burgard, W. & Fox, D. (2005) Probabilistic robotics. Cambridge, Massachusetts: The MIT Press.

Thrun, S., Fox, D., Burgard, W. & Dellaert, F. (2001) Robust Monte Carlo localization for mobile robots. Artificial Intelligence, 128(1–2), 99–141.

Thrun, S., Langford, J. & Verma, V. (2001) Risk sensitive particle filters. In: Advances in Neural Information Processing Systems (NIPS), vol 14, pp. 961–968.

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

Ueda, R., Arai, T., Sakamoto, K., Kikuchi, T. & Kaniwa, S. (2004) 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), pp. 2481–2486.

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

Wang, C.C., Thorpe, C., Thrun, S., Hebert, M. & Durrant-Whyte, H. (2007) Simultaneous localization and mapping in dynamic environments. Autonomous Robots, 19(1), 53–65.

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

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