To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2018, 1–20

Article

Simultaneous Pose and Reliability Estimation using Convolutional Neural Network and Rao-Blackwellized Particle Filter

Naoki Akai[a]* Luis Yoichi Morales[a], and Hiroshi Murase[a]

[a]Nagoya University, Furo, Chikusa, Nagoya, Japan

(*v1.0 released January 2018*)

In this study, we propose a novel localization approach that simultaneously estimates the reliability of estimation results. In the approach, a convolutional neural network (CNN) is used to make decision whether the localization process has failed or not. We train the CNN using a dataset that includes successful localization results and faults. However, the decision will contain some noise and many misdetection results may occur when the decision made by the CNN is used directly to detect faults. Therefore, we estimate both a robot's pose and reliability of the localization results based on the decision. To simultaneously estimate the robot's pose and reliability, we propose a new graphical model and implement a Rao-Blackwellized particle filter based on the model. We evaluated the proposed approach based on simulations and actual environments, which showed that the reliability estimated by the proposed approach can be used as an exact criterion for detecting localization faults. In addition, we show that the proposed approach can be applied in actual environments even when a dataset created from a simulation is used to train the CNN.

Keywords: Localization, Failure Detection, Reliability, Convolutional Neural Network, Rao-Blackwellized Particle Filter

1. Introduction

Localization is a fundamental technique employed in autonomous navigation. Many types of localization methods have been proposed and localization-based autonomous navigation systems have already been produced for robots and cars (e.g., [1, 2]). Guaranteeing the localization performance is very important for localization-based navigation because most of the functions in these systems use the localization results under the assumption that localization is always successful. However, this assumption is very dangerous because many unanticipated events can occur in real environments and they might lead to localization failures. In particular, a mobile robot must stop operating immediately when a localization failure occurs if the robot is navigating at high speed. In order to ensure safe operation in these conditions, it is useful to know the reliability of the localization results. In this study, we propose a novel localization approach that simultaneously estimates the robot's pose and the reliability of the localization results.

Figure 1 illustrates the concept employed in the proposed approach. We focus on two-dimensional (2D) laser scanner-based localization. Humans can easily recognize the faults in 2D laser scanner-based localization based on the localization image (top left) because mismatches between the scanned and map data are readily observed. The main idea of our approach is to find the mismatches using a convolutional neural network (CNN). We employ the CNN to model an assessment process that distinguishes whether localization has failed or not. However,

*Corresponding author. Email: akai@coi.nagoya-u.ac.jp

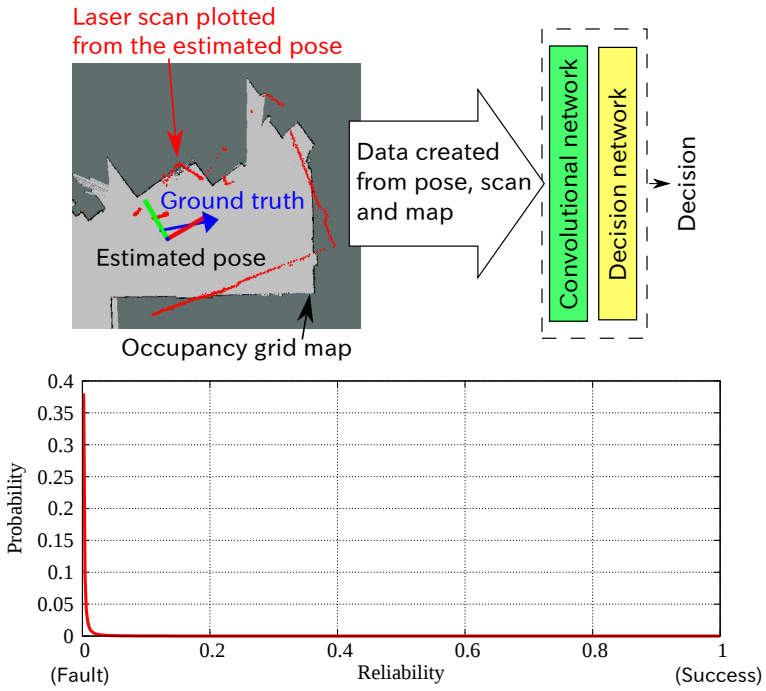

Figure 1. The failure of 2D laser scanner-based localization is easy to observe for a human. The input data for the CNN are created from an estimated pose, laser scan, and an occupancy grid map, and the CNN decides whether the localization process has failed. However, the decision will contain some noise and many misdetection results may occur when the decision made by the CNN is used directly to detect faults. The proposed approach simultaneously estimates a robot's pose and the reliability of the localization results using RBPF. The reliability is estimated based on the decision made by the CNN.

it must be noted that the output obtained from the CNN contains noise, so the reliability of the localization results should be estimated as a probabilistic variable instead of using the output directly. Therefore, we propose a new graphical model for simultaneously estimating a robot's pose and the reliability of the localization results.

This study makes three main contributions. The first contribution is that we propose the use of a CNN for detecting localization failures. The second contribution is the proposal of a novel graphical model for simultaneously estimating a robot's pose and the reliability of the localization results. Finally, we implemented a Rao–Blackwellized particle filter (RBPF) based on the graphical model in an actual mobile robot. The training dataset for the CNN is created using a simulation but it also operates successfully in actual implementations.

The remainder of this paper is organized as follows. Section 2 summarizes related work. In Section 3, we describe the proposed graphical model and discuss its advantages. In Section 4, we explain the models used to estimate the reliability of the localization results. In Section 5, we describe the implementation of the RBPF and CNN. In Section 6, we present evaluations of the proposed approach based on simulations and actual experiments. In Section 7, we give our conclusions.

2. Related works

In this section, we summarize model- and machine learning-based localization approaches.

2.1 *Model-based localization approaches*

Scan matching (SM) is widely used for mobile robot localization [3]. In SM, a cost function that models the matching of sensor observation and landmarks is used. Sensor pose on a given map is estimated by optimizing the cost function. The result obtained by SM does not estimate

uncertainty as it only performs optimization. Some approaches to estimate the uncertainty of the SM result have been proposed [4–6]. This uncertainty can be used as one of the criterion to detect a fault of the estimation, e.g., estimated results with large uncertainty are recognized as unreliable results. Some SM approaches that fuse other localization and SM estimation results by considering the estimated uncertainty have been proposed [7, 8]. By considering the uncertainty, the robustness of localization can be improved. However, this uncertainty does not provide explicit fault detection results of localization.

Monte Carlo localization (MCL) is a well-known algorithm for mobile robot localization [9]. This algorithm is based on a rigorous probabilistic process but it does not include a fault detection function. Thus, in order to detect faults, i.e., a kidnapped state [10], augmented MCL (AMCL[1]) method was proposed [10, 11]. To robustly perform localization using MCL in dynamic environments, selection scheme of reliable sensor measurements was proposed recently [12]. Our proposal estimates probability which represents whether localization has failed or not and this probability is regarded as reliability in the proposal.

Methods for recovering from a kidnapped state have also been proposed, which are called sensor resetting and expansion resetting [13, 14], and they use a similar approach to AMCL for detecting faults. It should be noted that we do not focus on a recovery method in this study, but instead we show that the reliability estimated using the proposed approach can be employed as an exact criterion for detecting faults.

AMCL algorithm estimates whether a kidnapped state occurs by observing short and long term likelihood histories and it determines how many random particles should be generated when a kidnapped state occurs. It may be difficult that AMCL successfully works in dynamic environments if the observation model does not consider dynamics of environment. In general, state space of particles should be extended to consider dynamics of environment, e.g., [15, 16]. In [15, 16], dynamic obstacles are added as hidden variables, and a map is also added as a hidden variable in [16]. By simultaneously estimating the robot's pose and dynamics of environment, stability of the localization can be improved. However, it should be noted that adding other hidden variables leads complexities of computation and algorithm.

Thrun *et al.* proposed a risk sensitive particle filter (RSPF) [17] that considers the risk estimated by a Markov decision process. In [17], the RSPF was applied to the mobile robot localization problem and it was shown that the RSPF allowed the robot to recover from a kidnapped state earlier than using the standard MCL in risky areas, e.g., the robot might collide with obstacles. Rapid recovery from a kidnapped state is possible using the RSPF but this method does not provide a fault detection function.

Sundvall *et al.* proposed a fault detection and recovery method for the localization problem [18] by using several sources for estimating the position of a robot, where it compares them to detect unanticipated deviations but without being deceived by drift or the different characteristics of positioning systems. Mendoza *et al.* also proposed a fault detection method [19], which uses redundant pose estimate information and it can achieve immediate automatic stopping to guarantee the safety of autonomous navigation. According to these previous studies, the use of redundant information is effective for detecting faults. The main difference of our approach and these approaches is that the proposed approach uses same information to detect faults. Thus, we do not need to use other positioning system for recognizing localization failure.

Improving the robustness of localization using sensor fusion or multi-hypotheses is important, as suggested by [20, 21]. However, it should be noted that many unanticipated events can occur in real environments so it is necessary to focus on the fault detection process. Fault detection and identification is a hot topic in the field of robotics and many studies have addressed this issue [22–24]. However, they focused mainly on mechanical faults and sensor failures such as breaks in actuators.

[1]Adaptive MCL which includes KLD-sampling is implemented on ROS and it is generally noted as AMCL, but we use AMCL to stand for augmented MCL in this paper.

The novelty of our proposed approach is that new hidden variable, called reliability, is added and output of the CNN, called decision, is used as an observable variable. Computation process of the approach is to be complex more than that of the standard MCL as new variables are added, however, it provides worth ability to detect localization failures. Our approach enables to give us explicit recognition results of localization faults.

2.2 *Machine learning-based localization approaches*

Machine learning algorithms have also been applied to the localization problem. Oore *et al.* proposed the use of a neural network (NN) for the localization problem [25]. In [25], measurement of several sonar sensors is used as input to the NN and likelihood distributions are predicted for updating probability distribution regarding the robot pose. Learning methods to select useful landmarks in environment have been proposed [26, 27]. These works are considered as pioneer works that applied machine learning approaches to the localization problem.

Loop-closure detection and place recognition are similar techniques to global localization and they can be utilized for fault detection. Bag of visual features-based methods [28] have been proposed, e.g., [29, 30]. However, manually designed features, such as SIFT [31], are generally used for detecting visual features. Machine learning-based relocalization approaches have also been proposed, such as pose localization using a regression forest and an RGB-D camera [32].

Recently, deep learning (DL)-based localization methods have been proposed, where DL can create effective features automatically [33], and a DL-based place recognition approach was developed that uses the features learned by a CNN [34]. In [35], the place recognition problem was solved as a classification problem using a CNN. Another DL-based relocalization approach was proposed by [36].

Choi *et al.* use a NN aided extended Kalman filter (EKF) SLAM [37]. The same approach was extended with the use of another type of NN [38]. Similar works are presented, e.g., [39], and the filter-based SLAM approach was enhanced using NNs. In addition, similar approaches, which integrate NNs to KF, were proposed to fuse INS and GPS, e.g., [40]. Alsayed *et al.* proposed a correction prediction method for 2D SLAM using ensemble multilayer perceptron [41]. These works improved localization and SLAM performance by using NNs, which predict suitable errors. On the other hand, our CNN makes decision whether localization has failed or not.

DL can also be employed to solve the time series state estimation problem recently. For example, DL algorithms were used to learn a generative model based on time sequence observations and action variables by [42], where they assumed that their proposed model is based on a KF-like model and the learning algorithm is referred to as deep KF. Haarnoja *et al.* proposed [43] the application of a DL algorithm that outputs observations and a covariance matrix for use as inputs by a KF. The result estimated by the KF is then utilized as a loss function for the network where this learning algorithm is referred to as backprop KF.

Alsayed *et al.* presented an interesting fault detection method for 2D LiDAR-based simultaneous localization and mapping (SLAM) [44]. They first considered fault cases of the 2D LiDAR-based SLAM and created the descriptor vectors and inference rules for the fault detection. Several machine learning methods were applied to the classification problem and they showed that approximately 85 % of classification accuracy could be obtained. Nobili *et al.* proposed a method for predicting alignment risk to prevent localization failure [45]. In the work, alignability is proposed while respecting localizability presented in [46] and the alignment risk is predicted based on support vector machine classifier using alignability and overlaps between target and source point clouds as input. In the global navigation satellite system (GNSS)-based localization system, there is a major problem called multipath. The detection of multipath can be considered as fault detection in the GNSS-based localization system. To detect multipath, Hsu applied machine learning approaches and showed that it is possible for the machine learning-based method to distinguish whether multipath is included or not in the received signals [47].

From a different perspective, Gaussian process (GP) [48] regression has been used to build

a vector signal map, e.g., WiFi and magnetic field [49, 50], and achieve localization. GP-based methods have also been employed to solve the vector field SLAM problem [51, 52]. These studies showed that GP can be applied to the localization problem and GP-based SLAM methods possess a function for solving the loop-closure problem.

As mentioned above, machine learning algorithms have generally been applied to solve the global localization, loop-closure detection, place recognition, time series state estimation, and fault detection problems. Our proposed approach uses a CNN for detecting faulty localization results. However, it might be expected that fault detection results using machine learning approaches are to be noisy if predicted results of the machine learnings are directly used. Our proposed approach employs RBPF that enables to reduce influence of noisy prediction results by the CNN. The use of RBPF is the novelty point of our approach.

3. Proposed method

3.1 *Definition of reliability*

According to the Advanced Product Quality Planning manual, reliability is defined as: "the probability that an item will continue to function at customer expectation levels at a measurement point, under specified environmental and duty cycle conditions" [53]. Therefore, we define reliability for the mobile robot localization problem as: "the probability that the error in the estimated pose is included within an acceptable region in order to perform the target task as expected." The region should be determined by considering the specific application. We briefly discuss the determination of this region in Section 5.

In the proposed approach, the value range for the reliability, r, ranges from 0 to 1. When the value is close to 1, it is considered that the localization error is included in the acceptable region. We estimate the probability distribution for r at the current time t.

3.2 *Graphical model and formulation*

It is assumed that we have a detector identifying faulty localization results. In this study, we use a CNN as the detector and it makes a decision denoted as d about whether the localization process has failed, where its value ranges from 0 to 1, and it is considered that the localization error is included in the acceptable region when the value is close to 1. However, we also consider that the decision will contain some noise and many misdetection results may occur when the decision made by the CNN is used directly to detect faults. Therefore, the reliability is estimated based on the decision.

Figure 2 illustrates the proposed graphical model, where there are two hidden and four observable variables in the model. The two hidden variables are the robot pose, $\mathbf{x}$, and the reliability of the localization results, r, which are depicted as white nodes. The four observable variables are the control input, $\mathbf{u}$, sensor observation, $\mathbf{z}$, map, $\mathbf{m}$, and the decision made by the CNN, d, which are depicted as gray nodes. This model can be considered as a general graphical model for localization when d and r are removed.

Our objective is to estimate the joint distribution for the hidden variables at time t, which is denoted as:

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

We apply the multiplication theorem to equation (1) and obtain the following equation:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) p(r_t | \mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}). \tag{2}$$

In this study, we implemented an RBPF based on the graphical model and we show that the

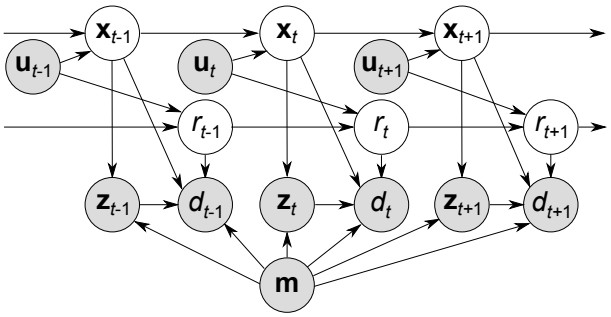

Figure 2. Graphical model for simultaneously estimating the current robot pose, $\mathbf{x}_t$, and the reliability of the estimation results, r_t. White and gray nodes denote hidden and observable variables, respectively. The CNN makes a decision using a sensor observation, $\mathbf{z}_t$, map, $\mathbf{m}$ and the pose. The reliability is considered as a hidden variable and is estimated using the decision made by the CNN, d_t, and the control input, $\mathbf{u}_t$.

joint distribution can be approximately estimated by the RBPF in the following.

We consider the first term of equation (2). We have two observable variables, $\mathbf{z}_t$ and d_t, for computing the likelihood of estimating the pose, we can apply Bayes' theorem twice and the first term is rewritten as:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) = \tag{3}$$
$$\eta p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) p(d_t|\mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t-1}),$$

where η is a normalization constant (η is always used as a normalization constant in this paper). It should be noted that we assume that the Markov property can be applied to the reliability estimation problem because the reliability is defined regard with localization result. The law of total probability is then applied to the second and third terms on the right-hand side of equation (3) and it can be rewritten as:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) = \tag{4}$$
$$\eta p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) \int p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t) dr_t$$
$$\int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}, \mathbf{m}, d_{1:t-1}) d\mathbf{x}_{t-1}.$$

According to this equation, we have two likelihood distributions, $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$ and $\int p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t) dr_t$, for determining the importance weight of the particles. The first is known as an observation model [10] and we show that the second can be computed analytically in Section 4. The distribution shown in equation (4) can be approximately estimated using a sampling-based method. Thus, an RBPF can be used to estimate the joint distribution of equation (1) if the second term of equation (2) is determined analytically.

Next, we focus on the second term. Bayes' theorem and the Markov property are first applied to the second term and it can be rewritten as:

$$p(r_t|\mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{m}, d_{1:t}) = \eta p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) p(r_t|\mathbf{x}_t, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, d_{1:t-1}). \tag{5}$$

We then apply the law of total probability to the second term on the right-hand side of equation (5) and we obtain the following equation:

$$\eta p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) \int p(r_t|r_{t-1}, \mathbf{u}_t) p(r_{t-1}|\mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}, d_{1:t-1}) dr_{t-1}, \tag{6}$$

where $p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})$ is a likelihood distribution for estimating the reliability and $p(r_t|r_{t-1}, \mathbf{u}_t)$

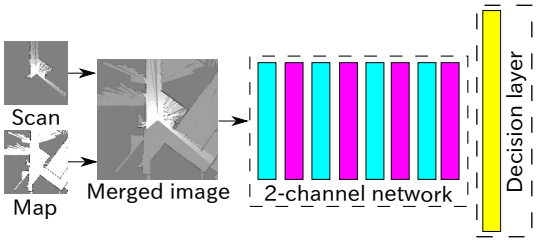

Figure 3. Configuration of the CNN. Two input images are merged and the merged image is treated as one image. Cyan, purple, and yellow layers denote the Conv2D+ReLU, max pooling, and fully connected layers, respectively. We use a slightly modified 2-channel model, which was originally proposed in [54].

is a distribution that represents the change in reliability relative to the movement of a robot. In general, the localization error will be increased by movement and it can be considered that the reliability will also decrease as a consequence of movement. Thus, we refer to $p(r_t|r_{t-1}, \mathbf{u}_t)$ as the "reliability decay model." In addition, $p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})$ is the likelihood distribution based on the decision, so we call it the "decision model." We describe these models in detail in Section 4.

3.3 *Advantages of simultaneous estimation*

The CNN can be applied directly to the localization results estimated using other approaches, e.g., standard MCL [9], and it can be employed as a fault detector. However, the simultaneous estimation process has several advantages.

As shown in equation (4), two likelihood distributions are used to compute the importance weights for the particles. In particular, the second term, $\int p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})p(r_t)dr_t$, reduces the influence of the noisy output from the CNN. If the CNN decides that the localization process is successful when the reliability value is low, then this value will be small. Thus, a particle with the correct pose, reliability, and decision will have a high likelihood in the proposed RBPF and the estimation performance will be more robust.

The CNN can be applied in environments where the learning data are not created. By contrast, other CNNs employed for relocalization, such as that proposed in [36], can only operate in environments where the learning data are created because the CNNs need to learn the relationships between the pose and images. Our proposed CNN can learn general localization failure phenomena, i.e., missmaches of the scan data and landmarks, based on successful localization and failures. These situations are similar even when the environment changes, so the CNN can operate in new environments encountered by a robot for the first time. Furthermore, highly accurate predictions are made by the CNN because the CNN does not need to recognize many complex features.

4. Decision and reliability decay models

4.1 *Decision model*

We use the CNN in order to decide whether the localization process has failed. Zagoruyko *et al.* proposed CNNs for predicting the similarity of an image pair [54]. The "2-channel model" CNN was proposed in [54] and we use a slightly modified 2-channel model to make decisions. Figure 3 illustrates how the model works in the proposed approach. First, two images are merged and the merged image is treated as one image. In the proposed approach, a scan image and a map image corresponding to an estimated pose are used as two images. The corresponding label, $y_i \in \{-1, 1\}$, was used by [54], so we use the same label. When the relationship is correct for the image pair, this means that localization is successful and $y_i = 1$ is set. When the localization process fails, we set $y_i = -1$. The value range for the output from the CNN, denoted as $d_t = f(\mathbf{x}_t, \mathbf{z}_t, \mathbf{m})$, is from -1 to 1, where $f(\cdot)$ is a mapping function modeled by the CNN.

We define the value range for the reliability from 0 to 1, so the output should be converted before building the decision model, as follows: $d_t \leftarrow (d_t + 1)/2$. In order to build the likelihood distribution based on the decisions, we use the beta distribution denoted as:

$$p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m}) = \frac{r_t^{a-1}(1-r_t)^{b-1}}{B(a,b)} \tag{7}$$
$$(a = d_t,\ b = 1 - d_t),$$

where $B(a, b)$ is the beta function.

The beta distribution has some advantages as a likelihood distribution with respect to our objective. The first advantage is that beta distribution is defined from 0 to 1, which allows us to easily apply the beta distribution to the reliability estimation problem. The second advantage is that the shape of the beta distribution is sharp around 0 or 1 when the values of a and b range from 0 to 1. The posterior distribution of the reliability will not be ambiguous with respect to success and failure, and thus faults can be detected as soon as possible.

4.2 *Reliability decay model*

In general, localization errors will increase as a consequence of movements by a robot so it is considered that the reliability of the localization results will also be decreased by movement. However, exactly modeling the relationship between movement and the degradation of localization reliability is not easy. Thus, we define the model heuristically as follows:

$$p(r_t|r_{t-1}, \mathbf{u}_t) = \begin{cases} \eta \exp(-(r_t - r_{t-1})^2/(2\sigma_m^2)) & (r_t \leq r_{t-1}) \\ 0 & (\text{otherwise}), \end{cases}$$

where σ_m^2 is the variance denoted as:

$$\sigma_m^2 = \alpha_1 \Delta d_t^2 + \alpha_2 \Delta \theta_t^2, \tag{8}$$

where $\mathbf{u}_t = (\Delta d_t, \Delta\theta_t)^T$ is the control input at time t, and α_1 and α_2 are constant values. This model indicates that the reliability must decrease where its degradation rate depends on the control input. Thus, we refer to this model as the reliability decay model. Figure 4 shows the effects of the model, where the distribution of the reliability is decreased slightly by applying the model. It should be noted that the motion model, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$, has effect to increase uncertainty of the pose and the uncertainty and reliability represent different things. Thus, we have two models regarding the robot motion in the proposed model.

By using equations (7) and (8), equation (6) can be determined analytically. Therefore, we can use the RBPF to estimate the joint distribution shown in equation (1).

5. Implementation

5.1 *RBPF*

A state of a particle, $\mathbf{s}$, comprises a 2D pose, $\mathbf{x} = (x, y, \theta)^T$, the probability distribution of the reliability, $p(r)$, and a weight, ω. We use the likelihood field model [55] as the observation model, $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$. $p(r)$ is approximately represented by a discrete probability distribution. In the initial state, the distribution is a uniform distribution. Estimation and re-sampling processes are performed after receiving data from a laser scanner and the update conditions are satisfied. The expectation value for the reliability of the maximum likelihood particle, $\int rp(r)dr$, is used

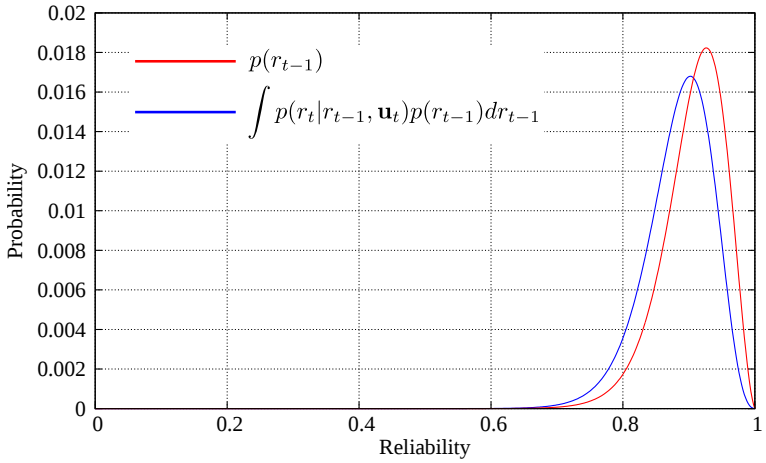

Figure 4. Effects of the reliability decay model.

as the reliability. The importance weight of the i-th particle at time t, ω_t^i, is computed as follows:

$$\omega_t^i = p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) \int p(d_t|r_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{m})p(r_t)dr_t\omega_{t-1}^i, \tag{9}$$

$$\omega_t^i \leftarrow \frac{\omega_t^i}{\sum_{j=1}^{M} w_t^j},$$

where M is the number of particles, which is set to 50. Pose updating based on the motion model, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$, is performed in parallel. α_1 and α_2 in equation (9) are determined experimentally, and we used $\alpha_1 = 0.002$ and $\alpha_2 = 0.004$ in this study. To decide whether performing resampling or not, effective sample size (ESS) [56] is used and it is denoted as:

$$ESS = \frac{1}{\sum_{i=1}^{M}(\omega^i)^2}, \tag{10}$$

Resampling is performed when following condition is satisfiled: $ESS < M/2$.

5.2 *CNN*

5.2.1 Dataset

The training dataset was created using a simulation. First, we built a 2D occupancy grid map in an actual environment using GMapping [56] with the robot shown in Fig. 5. After building the map, we navigated a simulated robot around the map in the simulation. Dynamic obstacles were also simulated in the map and laser scanning data were created. The ground truth could be obtained for the robot poses as we used a simulation. An image pair comprising a laser scan and grid map corresponding to the ground truth pose was used as a matching pair, and the label $y_i = 1$ was given. We randomly added noise to the ground truth pose to create a mismatched pair. When the difference in the position, $\sqrt{\Delta x^2 + \Delta y^2}$, or angle, $\Delta\theta$, between the ground truth and disturbed pose exceeded a threshold, an image pair comprising a laser scan and grid map corresponding to the disturbed pose was used as a mismatched pair, and the label $y_i = -1$ was given.

As mentioned in Section 3, localization faults should be decided by considering the specific application. Our main objective is autonomous navigation, so the main focus is position tracking in the localization problem. Thus, we aim to detect localization faults as soon as possible and

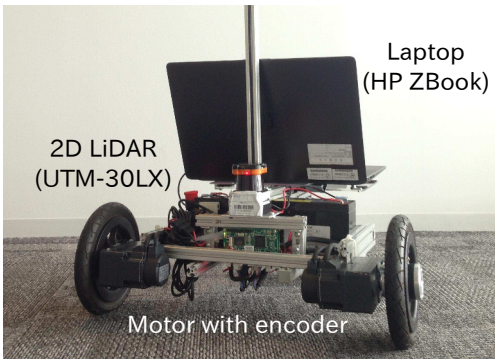

Figure 5. Experimental platform.

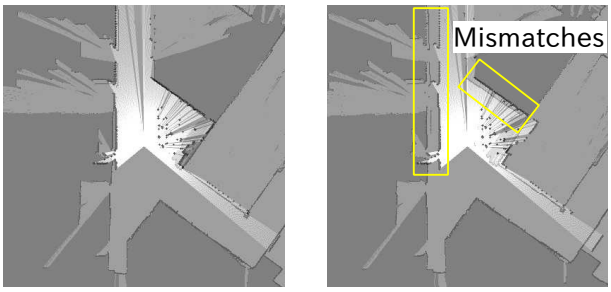

Figure 6. Example of an input image pair. Note that these images are transparent to ensure visibility but the actual input data are not transparent.

try to recover from the fault. When we use MCL for the position tracking, recovering from the fault can be performed easily by generating particles around the current estimated pose if the localization error is small. However, we must consider that mismatches might occur when particles are generated over a wide area. A sampling range of approximately 0.5 m and 3 degrees is suitable for recovering from a small fault without mismatches in our experience. Thus, the threshold values for differences in the position and angle when deciding faults were set to 0.5 m and 3 degrees, respectively. Figure 6 shows an example of an image pair for successful and failed localization. In the actual implementation, the resolution and size of the image were set to 0.05 m and 20 m, respectively.

5.2.2 Network configuration

We used Keras as development environment for the CNN [57]. The network developed in this study received an image measuring 100×100 and the output was a single value. There were four Conv2D+ReLU and max pooling layers, and one fully connected layer between the input and output layers, as shown in Fig. 3. Hyperbolic tangent was used as the activation function in the output layer.

We trained the CNN model in a supervised manner. A squared error-based loss term and squared $l2$-norm regularization were used in the objective function:

$$\min \frac{\lambda}{2}||\omega^{\text{net}}||_2 + \sum_{i=1}^{N}(y_i - d_i)^2, \tag{11}$$

where ω^{net} are the weights of the network, d_i is the network output for the i-th training sample, N is the number of training data, and $y_i \in \{-1, 1\}$ is the corresponding label. The weights were initialized randomly with a normal distribution and Adadelta [58] was used as the optimizer.

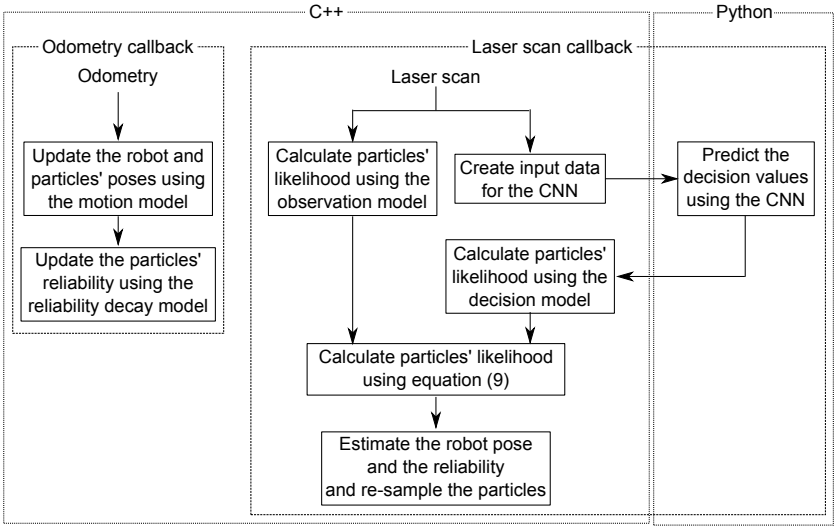

Figure 7. Computation flow of the proposed approach.

6. Evaluation

6.1 *Conditions*

6.1.1 Experimental platform

The experimental platform is shown in Fig. 5. A 2D LiDAR (UTM-30LX) was mounted on the robot. The measurement range, angle, and resolution of the LiDAR system were 30 m, 270 degrees, and 0.25 degrees, respectively. In the simulation, we reproduced the same specification as that for the LiDAR. A laptop (HP ZBook) was used as the main computer and a GPU was installed. The computation of the CNN was performed on the GPU. Intel(R) Xeon(R) CPU E3-1545M v5@2.90 GHz and NVIDIA Corporation GM107GLM (Quadro M1000M) are respectively used as CPU and GPU.

6.1.2 Computation flow

We implemented the proposed approach using ROS architecture and Fig. 7 depicts the computation flow. The callback function of the laser scan data parallelly runs to calculate particles' likelihood with the use of both the observation and decision models. The input data for the CNN is created in the C++ node and the CNN is implemented as the python node. After predicting the decision values using the CNN, the particles' likelihood are calculated using equation (9) in the C++ node. We measured computation time from the start of the laser scan callback to the likelihood calculation using the above mentioned platform.

6.1.3 Learning data and experimental environments

We used four indoor environments to create the learning dataset, where approximately 5000 success and failure image pairs were created. The experiments were conducted in environments where the learning data were not created. Figure 8 shows the occupancy grid maps for the experimental environments.

6.2 *Performance of the CNN*

Figure 9 shows the receiver operating characteristic (ROC) curves computed for each test dataset created in experimental environments (a) and (b). We also created a test dataset using actual sensor data (the pose estimated by MCL was used as the ground truth). The results were classified as successful localization when the value predicted by the CNN exceeded 0.5, whereas

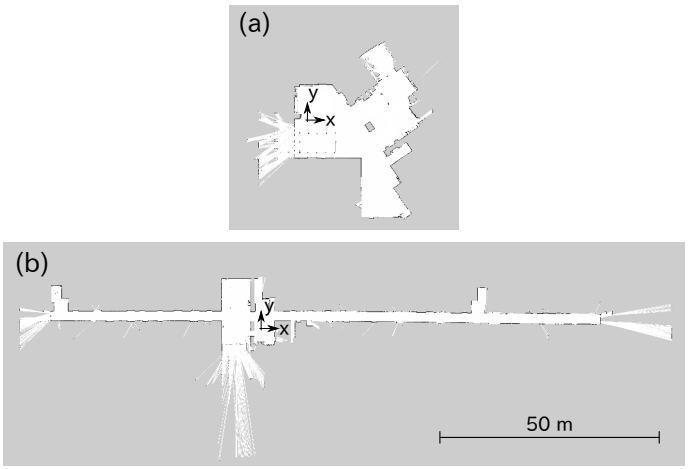

Figure 8. Occupancy grid maps for the experimental environments.

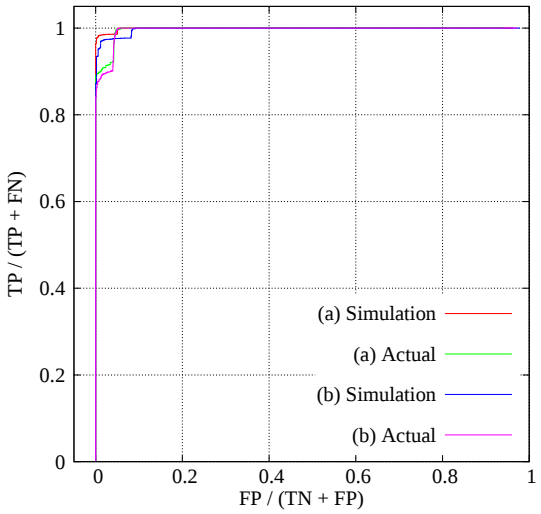

Figure 9. ROC curves. TP, FP, FN, and TN denote true positive, false positive, false negative, and true negative, respectively.

other results were classified as failures. Note that value of the CNN's output was converted from 0 to 1. As shown in the figure, the CNN could make accurate decisions using the dataset created in the simulation and actual environments.

6.3 *Method used for comparison*

In AMCL, the random particle rate (RPR) represents how many random particles need to be generated to recover from the kidnapped state, and it is computed as follows:

$$\omega_{\text{slow}} \leftarrow \alpha_{\text{slow}}(\omega_{\text{ave}} - \omega_{\text{slow}}), \tag{12}$$

$$\omega_{\text{fast}} \leftarrow \alpha_{\text{fast}}(\omega_{\text{ave}} - \omega_{\text{fast}}), \tag{13}$$

$$\text{RPR} = \max(0, 1 - \omega_{\text{fast}}/\omega_{\text{slow}}), \tag{14}$$

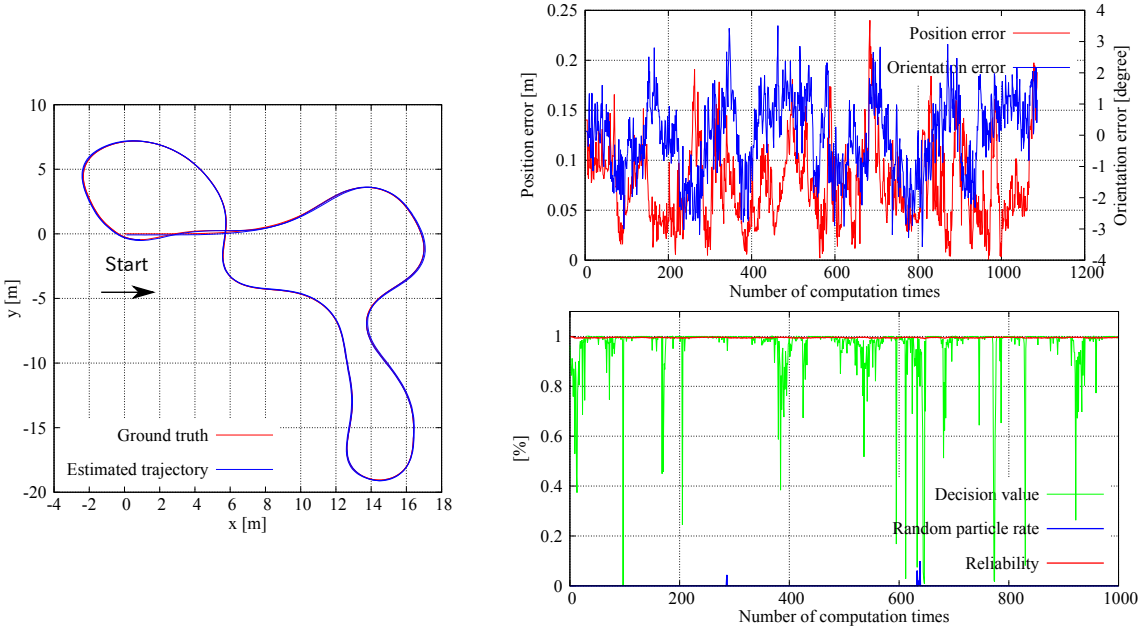

Figure 10. The simulation result with the successful robot pose tracking results.

where α_{slow} and α_{fast} $(0 \leq \alpha_{\text{slow}} \ll \alpha_{\text{fast}} \leq 1)$ are constant values, and ω_{ave} is defined as:

$$\omega_{\text{ave}} = \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{z}_t | \mathbf{x}_t^i, \mathbf{m}) \omega_{t-1}^i. \tag{15}$$

When the RPR exceeds 0, it may be considered that the kidnapped state occurs. We compared the reliability estimated using the proposed approach, r, and the RPR in the simulation experiments. It should be noted that ω_{slow} and ω_{fast} were reset to ω_{ave} to avoid spiraling off into complete randomness when the RPR exceeded 0. AMCL was implemented using ROS and with the values recommended in [59]: $\alpha_{\text{slow}} = 0.001$ and $\alpha_{\text{fast}} = 0.1$. We also used the same values in the experiments.

6.4 *Simulation experiments*

We conducted simulation experiments in the environment shown in Fig. 8(a). It should be noted that moving obstacles were simulated, so the experiments were conducted in the dynamic environment. Fig. 10, 11, and 12 show the results, where the figure panels at the left represent the trajectories for the ground truth (red) and those estimated using the proposed approach (blue). The figure panels at the top right show the position and orientation errors (red and blue) and the figure panels at the bottom right show the reliability (red), the value predicted by the CNN for the maximum likelihood particle (green), and the RPR (blue).

In case Fig. 10, the robot went round the same path twice and our approach successfully tracked the robot's pose during the navigation process. The decision value was very close to zero on some occasions but the reliability was always close to 1. This result shows that the influence of the noisy prediction results obtained by the CNN could be reduced by the proposed RBPF.

In case Fig. 11, we started the experiment with an incorrectly estimated pose. The particles were spread according to the motion model and the localization result converged to the correct pose during the navigation process. At point A, the reliability was close to 1 and it remained close to 1 after this point.

In case Fig. 12, we conducted the experiment with a small amount of motion noise, i.e.,

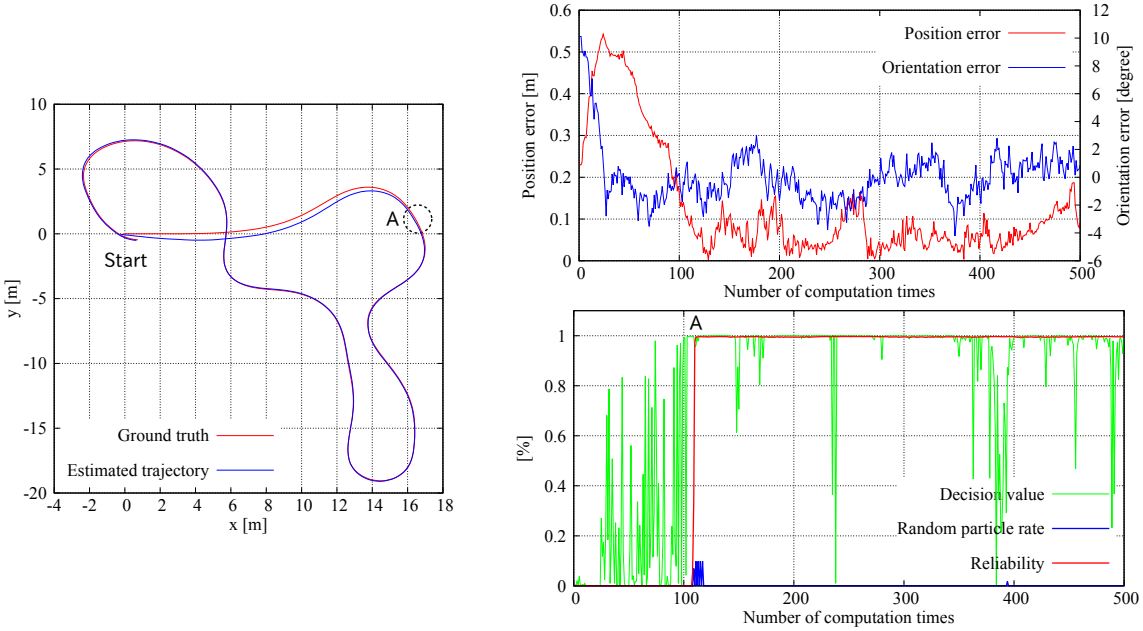

Figure 11. The simulation result with the incorrect initial pose.

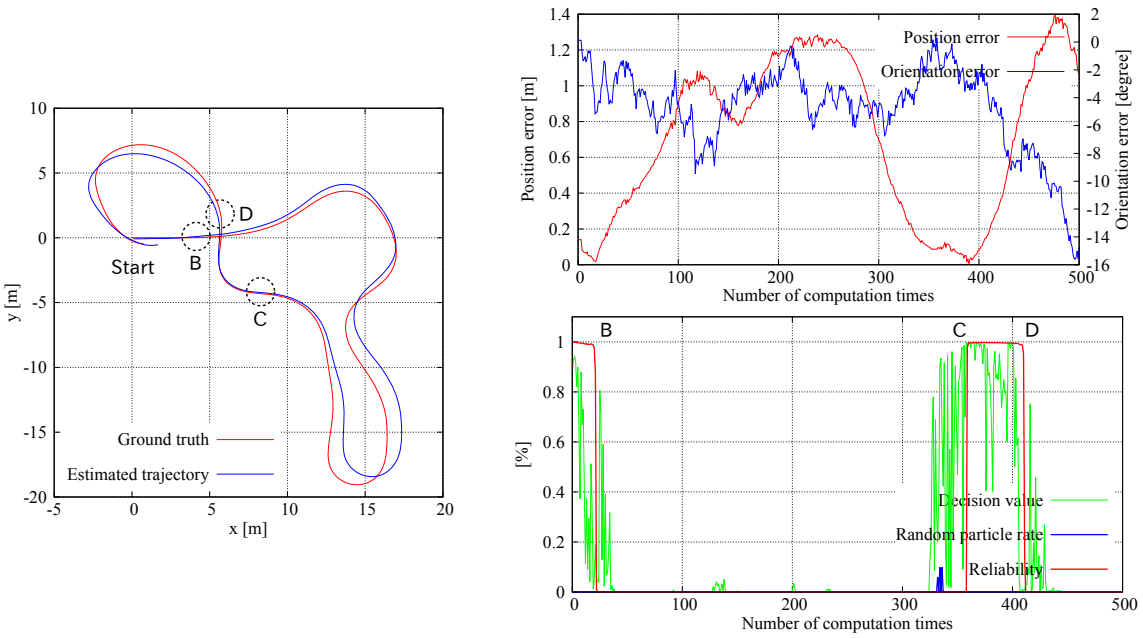

Figure 12. The simulation result with the incorrect localization parameters.

incorrect localization parameters. The particles could not be spread even when the robot moved, so the estimated result could not track the robot's pose. The reliability was close to zero at point B. The localization error decreased around point C and the reliability was 1. However, the localization error increased again and the reliability was close to zero around point D. These results show that the reliability exactly described the success and failure of the localization results.

In each case, the RPR sometimes exceeded zero (see blue lines of each bottom right figure) and some incorrect decisions were made. For example in case Fig. 10, the RPR exceeded 0 even the localization error was always small. In case Fig. 12, the RPR did not exceed 0 even when the position error exceeded 1 m around the computational time step of 100. According to these

Table 1. Average and standard deviation of computation times for calculating likelihood (in millisecond).

# particles	50	100	150	200
Standard MCL	1.65 (0.47)	2.95 (0.42)	4.27 (0.52)	6.69 (1.03)
Proposed approach	122.59 (2.39)	229.31 (2.69)	374.05 (17.77)	562.94 (28.06)

results, the reliability estimated by our approach can be used as an exact criterion to detect localization failures.

Table 1 shows a comparison result of computation times. In the result, likelihood calculation time of the standard MCL and proposed approach was measured. The computation time of the standard MCL is significantly smaller than that of the proposed approach. The most long part of taking time is the prediction part using the CNN and it has over 90 % of times. We aim to accelerate the computational time, especially the CNN computation, as the future work.

We conducted similar experiments in different simulation environments and obtain similar results described above. To show the performance, we uploaded a video showing the simulation experiments to the Internet and it is freely available[1]. Robustness and effectiveness of the approach in highly dynamic environments can be confirmed from the video.

6.5 *Real indoor experiment*

We conducted a real indoor experiment in the environment shown in Fig. 8(a) without further learning using real environment data. In this experiment, we manually moved the robot to the kidnapped state during the navigation process. The particles were regenerated around the current estimated pose when the reliability was less than 0.95 for recovering the kidnapped state. It should be noted that we did not use recovery methods such as those proposed by [13, 14], and the particles were simply randomly regenerated.

Figure 13 shows the estimated trajectory and reliability. This environment was a garage containing cars and five cars were parked in the experiment. The five cars were not mapped in the used occupancy grid map. We kidnapped the robot at points A, B, and C. The reliability decreased at these three points and the particles were regenerated. Thus, the localization result was recovered immediately from the kidnapped state. We also uploaded a video showing the experiment to the Internet and it is freely available[2].

Through the simulation and real indoor experiments, we showed that the reliability was exactly estimated even though environments are not static. Especially, robustness of the approach in highly dynamic environments was confirmed through the simulation experiments. Therefore, we conclude that the reliability could be used as an exact criterion for detecting localization failures.

6.6 *Discussion*

Figure 14 shows the output results from the final convolution layer obtained using guided Grad-CAM [60]. Guided Grad-CAM tells us region in which convolution layers of the CNN are responded. These image pairs were created using actual sensor data. The figures at the top and bottom show successes and failures, respectively, where the light green regions indicate that the filters in the final convolution layer obtained strong reactions. Figure 14 shows that the CNN focused on the regions around landmarks. In particular, the landmarks that did not match with the laser scan were focused on well in both the successes and failures. The CNN could have learned the mismatches with the laser scan and the landmarks.

We used approximately 5000 success and failure image pairs to train the CNN. This number might be considered small compared with the numbers required by other CNNs. However, our

[1] https://www.youtube.com/watch?v=n_3BXbCpYBk&t=1s
[2] https://www.youtube.com/watch?v=QzG3beQkQnY

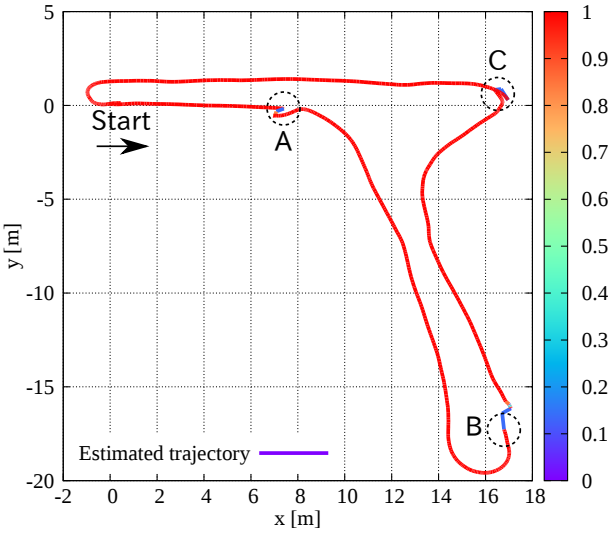

Figure 13. Results obtained in the real indoor experiment. The robot was kidnapped at points A, B, and C. The particles were regenerated when the reliability was less than 0.95. The kidnapped state was detected well by our approach and faults were recovered.

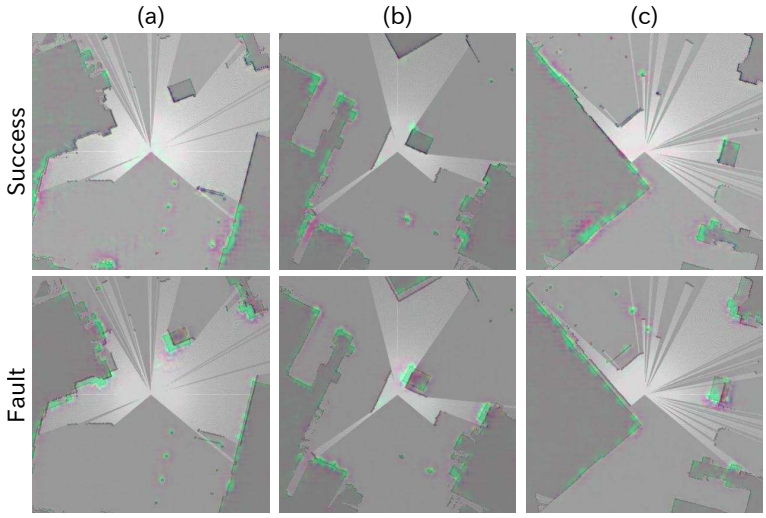

Figure 14. Visualization of the output results from the final convolution layer obtained using guided Grad-CAM [60]. Light green regions indicate that the filters in the final layer obtained strong reactions.

method uses simple input data and the CNN learns the simple relationships between the scan and the map, so it does not require a large amount of images. This is a key efficiency of the proposed approach, which combines model- and learning-based techniques.

The resolution of the input image is an important factor that affects accurate decision making by the CNN. However, the computational time required will be problematic when a large resolution image is used. A laser scanner with a wide range is generally used to perform localization in outdoor environments, especially for autonomous cars' localization. Thus, it might be necessary to create the input data when the proposed approach is used for outdoor localization with some devices. For example, omitting the free space where the laser beams passed through might be considered effective for some devices because the CNN could learn the mismatches between the scan and map. However, we need to consider how to create meaningful data even if the geometrical consistency is lost when omitting the free space.

Through the experiments, we showed that the CNN could have superior performance as the fault detector. On the other hand, our approach can employ another type of fault detectors, for example, a model-based fault detector. To create a describable localization system, a model-based

fault detector should be used. However, it is not easy to model localization failures. Therefore, we used the CNN as the fault detector in this study. Discussing models of localization failures is still significant and we are considering that the CNN's response as shown in Fig. 14 might be utilized to create such models.

7. Conclusion

In this study, we proposed a novel localization approach that simultaneously estimates a robot's pose and the reliability of the localization results. We proposed a new graphical model and implemented an RBPF based on the model for simultaneous estimation. In the proposed method, a CNN is used to decide whether localization has failed and the reliability is estimated based on this decision. By applying the simultaneous estimation, influence of noisy prediction results by the CNN could be reduced. Our experiments demonstrated that the reliability can be used as an exact criterion for detecting localization failures.

In the future, we will apply our approach to autonomous cars. We aim to accelerate the computational time, especially the GPU computation, in order to estimate the reliability of autonomous cars at rapid speeds. Creating a model of localization failure is also the future work for developing a describable localization system using the proposed graphical model.

ACKNOWLEDGMENT

This research was supported by the Center of Innovation Program (Nagoya-COI; Mobility Society leading to an Active and Joyful Life for Elderly) funded by the Japan Science and Technology Agency and Artificial Intelligence Research Promotion Foundation. We also greatly appreciate Prof. Eijiro Takeuchi's advice.

References

[1] L.Y. Morales, A. Carballo, E. Takeuchi, A. Aburadani, and T. Tsubouchi. "Autonomous robot navigation in outdoor cluttered pedestrian walkways," *Journal of Field Robotics (JFR)*, vol. 26, no. 8, pp. 609–635, 2009.

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

[3] P.J. Besl and H.D. McKay. "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, no. 2, 1992.

[4] O. Bengtsson and A. J. Baerveldt. "Localization in changing environments - estimation of a covariance matrix for the IDC algorithm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[5] A. Censi. "An accurate closed-form estimate of ICP's covariance," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[6] E.B. Olson. "Real-time correlative scan matching," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[7] Y. Hara, S. Bando, T. Tsubouchi, A. Oshima, I. Kitahara, and Y. Kameda. "6DOF iterative closest point matching considering a priori with maximum a posteriori estimation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

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

[9] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. "Monte Carlo localization for mobile robots," in

Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), vol. 2, pp. 1322–1328, 1999.

[10] S. Thrun, W. Burgard, and D. Fox. "Probabilistic robotics," *MIT Press*, 2005.

[11] J. Gutmann and D. Fox. "An experimental comparison of localization methods continued," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 454–459, 2002.

[12] J. Kim and W. Chung. "Robust localization of mobile robots considering reliability of LiDAR measurements," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6491–6496, 2018.

[13] S. Lenser and M. Veloso. "Sensor resetting localization for poorly modelled mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1225–1232, 2000.

[14] R. Ueda, T. Arai, K. Sakamoto, T. Kikuchi, and S. Kamiya. "Expansion resetting for recovery from fatal error in Monte Carlo localization - comparison with sensor resetting methods," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[15] M. Montemerlo, S. Thrun, and W. Whittaker. "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[16] C.C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research (IJRR)*, vol. 26, no. 9, pp. 889–916, 2007.

[17] S. Thrun, J. Langford, and V. Verma. "Risk sensitive particle filters," *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[18] P. Sundvall and P. Jensfelt. "Fault detection for mobile robots using redundant positioning systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3781–3786, 2006.

[19] J.P. Mendoza, M. Veloso, and R. Simmons. "Mobile robot fault detection based on redundant information statistics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[20] L. Marín, M. Vallés, Á. Soriano, Á. Valera, and P. Albertos. "Multi sensor fusion framework for indoor-outdoor localization of limited resource mobile robots," *Sensors*, pp. 14133–14160, 2013.

[21] G. Jochmann, S. Kerner, S. Tasse, and O. Urbann. Efficient multi-hypotheses unscented Kalman filtering for robust localization *RoboCup 2011: Robot Soccer World Cup XV*, pp. 222–233, 2011.

[22] P. Goel, G. Dedeoglu, S.I. Roumeliotis, and G.S. Sukhatme. "Fault detection and identification in a mobile robot using multiple model estimation and neural network," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[23] V. Verma, G. Gordon, R. Simmons, and S. Thrun. "Particle filters for rover fault diagnosis," *Robotics & Automation Magazine special issue on human centered robotics and dependability*, 2004.

[24] D. Stavrou, D.G. Eliades, C.G. Panayiotou, and M.M. Polycarpou. "Fault detection for service mobile robots using model-based method," *Autonomous Robots*, vol. 40, no. 2, pp. 383–394, 2016.

[25] S. Oore, G.E. Hinton, and G. Dudek. "A mobile robot that learns its place," *Neural Computation*, vol. 9, pp. 683–699, 1997.

[26] R. Greiner and R. Isukapalli. "Learning to select useful landmarks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, no. 3, pp. 437–49, 1996.

[27] S. Thrun. "Bayesian landmark learning for mobile robot localization," *Machine Learning*, vol. 33, no. 1, pp. 41–76, 1998.

[28] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray. "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22, 2004.

[29] A. Angeli, D. Filliat, S. Doncieux, and J. Meyer. "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 5, pp. 1027–1037, 2008.

[30] D.G. López and J.D. Tardos. "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics (TRO)*, vol. 28, no. 5, pp. 1188–1197, 2012.

[31] D.G. Lowe. "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[32] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. "Scene coordinate regression forests for camera relocalization in RGB-D images." in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), pp. 2930–2937, 2013.
[33] A. Krizhevsky, I. Sutskever, and G.E. Hinton. "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NIPS)*, 2012.
[34] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. "On the performance of ConvNet features for place recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
[35] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. "Deep learning features at scale for visual place recognition," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
[36] A. Kendall, M. Grimes, and R. Cipolla. "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946, 2015.
[37] M. Choi, R. Sakthivel, and W.K. Chung. "Neural network-aided extended Kalman filter for SLAM problem," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1686–1690, 2007.
[38] K.Yu.K. Alexandr and M.A. Sobolev. "Recurrent neural network and extended Kalman filter in SLAM problem," *IFAC Proceedings Volumes*, vol. 46, no. 20, pp. 23–26, 2013.
[39] K.-S. Choi and S.-G. Lee. "Enhanced SLAM for a mobile robot using extended Kalman filter and neural networks," *International Journal of Precision Engineering and Manufacturing*, vol. 11, no. 2, pp. 255–26, 2010.
[40] R. Sharaf, A. Noureldin, A. Osman, and N. El-Sheimy. "Online INS/GPS integration with a radial basis function neural network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 3, pp. 8–14, 2005.
[41] Z. Alsayed, G. Bresson, A. Verroust-Blondet, and F. Nashashibi. 2D SLAM correction prediction in large scale urban environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5167–5174, 2018.
[42] R.G. Krishnan, U. Shalit, and D. Sontag. "Deep Kalman filters," *arXiv:1511.05121*, 2015.
[43] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel. "Backprop KF: Learning discriminative deterministic state estimators," *Advances in Neural Information Processing Systems (NIPS)*, 2016.
[44] Z. Alsayed, G. Bresson, A, Verroust-Blondet, and F. Nashashibi. "Failure detection for laser-based SLAM in urban and peri-urban environments," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 126–132, 2017.
[45] S. Nobili, G. Tinchev, and M. Fallon. "Predicting alignment risk to prevent localization failure," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1003–1010, 2018.
[46] W. Zhen, S. Zeng, and S. Scherer. "Robust localization and localizability estimation with a rotating laser scanner," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6240–6245, 2017.
[47] L.T. Hsu. "GNSS mulitpath detection using a machine learning approach," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1414–1419, 2017.
[48] C.E. Rasmussen and C.K.I. Williams. "Gaussian processes for machine learning," *MIT Press*, 2006.
[49] B. Ferris, D. Hahnel, and D. Fox. "Gaussian processes for signal strength-based location estimation," *Robotics: Science and Systems (RSS)*, 2007.
[50] N. Akai and K. Ozaki. "Gaussian processes for magnetic map-based localization in large-scale indoor environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4459–4464, 2015.
[51] B. Ferris, D. Fox, and N. Lawrence. "WiFi-SLAM using Gaussian process latent variable models," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2480–2485, 2007.
[52] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning. "Simultaneous localization and mapping using ambient magnetic field," in *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2010.
[53] Q. Xin. "Diesel engine system design," *Woodhead Publishing in Mechanical Engineering*, p. 42, 2011.
[54] S. Zagoruyko and N. Komodakis. "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4353–4361, 2015.
[55] R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard. "Monte Carlo localization in outdoor terrains

using multilevel surface maps," *Journal of Field Robotics (JFR)*, vol. 25, vol. 6–7, pp. 346–359, 2008.
[56] G. Grisetti, C. Stachniss, and W. Burgard. "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics (TRO)*, vol. 23, no. 1, pp. 34–46, 2007.
[57] F. Chollet *et al.* "Keras," *GitHub*, 2015. [Online]. Available: `https://github.com/fchollet/keras`
[58] H.B. McMahan and M. Streeter. "Delay-tolerant algorithms for asynchronous distributed online learning," *Advances in Neural Information Processing Systems (NIPS)*, 2014.
[59] [Online]. Available: `http://wiki.ros.org/amcl`
[60] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. "Grad-CAM: visual explanations from deep networks via gradient-based localization," *arXiv:1610.02391*, 2016.