Misalignment Recognition Using Markov Random Fields with Fully Connected Latent Variables for Detecting Localization Failures

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

***Abstract*—Recognizing misalignment between sensor measurements and objects that exist on a map due to inaccuracies in localization estimation is challenging. This can be attributed to the fact that the sensor measurements are individually modelled for solving the localization problem, resulting in entire relations of the measurements being ignored. This paper proposes a misalignment recognition method using Markov random fields with fully connected latent variables for the detection of localization failures. The proposed method estimates the classes of each sensor measurement that are aligned, misaligned, and obtained from unknown obstacles. The full connection allows us to consider the entire relation of the measurements. A misalignment can be exactly recognized even when partial sensor measurements overlap with mapped objects. Based on the class estimation results, we are able to distinguish whether the localization has failed or not. The proposed method was compared with six alternative methods, including a convolutional neural network, using datasets composed of success and failure localization samples. Experimental results show that classification accuracy of the localization samples using the proposed method exceeds 95 % and outperforms the other examined methods.**

***Index Terms*—Localization, Failure Detection and Recovery, Probability and Statistical Methods.**

I. Introduction

We GENERALLY assume that sensor measurements are independent in the localization problem [1]. This assumption allows us to simply model the sensor measurements because it is not necessary to model how entire sensor measurements are simultaneously obtained. The probabilistic distribution of the measurements is denoted as:

$$p(\mathbf{Y}|\mathbf{x},\mathbf{m}) = \prod_{k=1}^{K} p(\mathbf{y}_k|\mathbf{x},\mathbf{m}), \tag{1}$$

where $\mathbf{x}$ is a target pose, $\mathbf{m}$ is a map, and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K)^T$ are the sensor measurements. This model is referred to the measurement model. Features detected in the sensor measurements, e.g., lines, can also be used as $\mathbf{Y}$ [2].

This assumption, however, causes a problem whereby the entire relation of the sensor measurements is ignored. Consequently, it is hard to recognize misalignment that has occurred because of localization failure. Moreover, misalignment means that the sensor measurements are not be correctly matched to the corresponding objects on a map, for example landmarks. Map matching [3] is an approach used to consider the entire relation, but it is unable to consider physical measurement noises [1]. Other approaches to overcome the problem include simultaneously estimating environment dynamics [4], [5], but such simultaneous estimation cannot be dealt with by the localization framework. We therefore need to use the independent assumption to practically solve the localization problem.

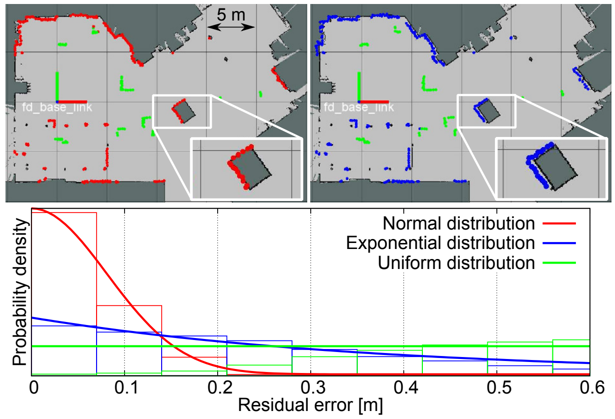

Fig. 1. Success (top left) and failure (top right) localization samples with the sensor measurement class estimation by the proposed method; aligned (red), misaligned (blue), and unknown (green). To estimate the classes, we utilize phenomena whereby the residual errors of the classes are in accordance with normal, exponential, and uniform distributions as shown in the bottom graph.

In this paper, we propose a misalignment recognition method that uses Markov random fields (MRFs) with fully connected latent variables (FCLVs). The full connection allows us to consider the entire relation of the sensor measurements without performing complex measurement modelling. The top of Fig. 1 illustrates the process of the proposed method. The proposed method divides the sensor measurements into three classes; aligned (red), misaligned (blue), and unknown (green). The measurement classes depicted in the top of Fig. 1 were estimated using the proposed method. As can be seen from the top right of Fig. 1, the misalignment can be exactly recognized even though the partial sensor measurements are overlapping with the landmarks. To recognize the measurement classes, we utilize phenomena where residual errors of the classes are in accordance with normal, exponential, and uniform distributions, respectively. The bottom of Fig. 1 shows the histograms relating to the residual errors and the distributions.

Manuscript received: February 24, 2019; Revised May, 20, 2019; Accepted June, 18, 2019.

This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments.

[1]N. Akai and H. Murase are with the Graduate School of Informatics, Nagoya University, Nagoya 464-8603, Japan akai@nagoya-u.jp

[2]L. Y. Morales and T. Hirayama are with the Institute of Innovation for Future Society, Nagoya University, Nagoya 464-8601, Japan

The histograms were established from the datasets used in this work. The dataset is presented in further detail in Section IV.

The main contribution of this work is to propose the use of MRFs with FCLVs in recognizing a misalignment. Additionally, we describe how to detect localization failures based on the class estimation. Classification performance of the proposed method for success and failure localization samples was compared to six existing methods including a convolutional neural network (CNN). Experimental results show that classification accuracy using the proposed method exceeded 95 % and outperformed the other methods.

The rest of this paper is organized as follows. Section II summarizes related works. Section III details the proposed method. Section IV and V describe the experimental conditions and results. Section VI concludes this work.

II. Related works

As mentioned in Section I, the sensor measurements are independently modelled in localization approaches, for example in Kalman and particle filters (PF) [1]. Consequently, it is hard to recognize a misalignment based on the measurement model. Gutmann *et al*. proposed a failure detection method for PF-based localization that observes history of likelihoods [6]. The method does not explicitly recognize the misalignment.

In optimization-based localization approaches, for example ICP and NDT scan matchings [7], [8], corresponding points (or features) are searched for individually from the map. The cost function is then calculated using the corresponding pairs and the relative pose that minimizes the cost function is estimated. Thus, the entire relation of the measurements is also ignored.

In [9], [10], the definition of map brokenness and an algorithm to measure inconsistencies of the map are presented. However, this method has a severe limitation in autonomous navigation because it is dependent on a given reference map. A popular method to detect the failure of point registration is to use distances between corresponding points, for example root mean square, as presented in [11]. The partitioned mean normals measure that uses consistency between normals is presented in [12]. However, insufficient performance of classification methods that are based on thresholds of corresponding pairs when registration errors are small is presented in [13].

Some authors propose failure detection methods that use a redundant position estimation system [14], [15]. Localization results that might be incorrect can be detected based on the redundant information. However, these methods do not give us the criterion to detect failures in each estimation.

Recently, a number of authors have proposed localization failure detection methods based on machine learning (ML) algorithms [16], [17]. The ML-based methods allow us to implicitly model complex relationships. The ML-based methods have also been applied in the detection of multipath, which causes failure in GNSS-based localization [18]. Almqvist *et al*. compared several misaligned point cloud detection methods including the ML-based methods [19] and showed that AdaBoost [20] executed a better overall performance.

Deep neural networks have been applied to solve many problems. End-to-end localization [21] and rigid transformation estimation [22] have been proposed. It might be suggested that the networks learn the entire relation of source and target data, because the process to find corresponding points and/or features between source and target data is omitted.

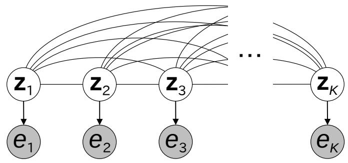

Fig. 2. The graphical model of the MRFs with the FCLVs.

We previously proposed a localization method using the measurement model that explicitly considers environment dynamics and showed that the robustness could be improved [23]. Even if environmental dynamics are explicitly considered, localization failure detection cannot be achieved. We also previously proposed a CNN-based localization failure detector in [24], [25]. The CNN-based detector distinguishes, exactly, whether localization has failed or not. In this paper, we compare the proposed method with the detector and show that the proposed method outperforms the detector.

Similar methods to our method are presented in [26]–[30]. In [26], the use of conditional random fields (CRFs) with FCLVs in recognizing in and outliers between two point clouds is presented. The method enables a capacity to ignore outliers when computing the cost function and the alignment robustness can thus be increased. In [27], the use of the CRFs to assess map quality is presented. The CRFs can recognize suspicious mapping results, for example, when the same objects are not correctly mapped as the same landmarks. The method is quite similar to our method, because both methods can recognize the misalignment; however, our method uses MRFs with FCLVs for misalignment recognition. In particular, our method does not require a training dataset. In [28], ICP scan matching using MRFs without FCLVs is presented. In the method, MRFs are also used to recognize in and outliers between two point clouds. Through a series of experiments, we show that the use of FCLVs yields better recognition performance. Furthermore, we estimate the localization failure probability based on the misalignment recognition. In [29], [30], residual error distribution modelling for calculating likelihood is presented. Our method also utilizes residual error distributions, but with a differing objective. We utilize the distributions to estimate the measurement classes.

III. Proposed method

Figure 2 illustrates a graphical model of the MRFs with the FCLVs. $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_K)^T$ with $\mathbf{z}_k \in \mathcal{R}^3$, $z_{k,l} \in \{0, 1\}$, and $\sum_{l=1}^{3} z_{k,l} = 1$ are the latent variables, $\mathbf{e} = (e_1, e_2, ..., e_K)^T$ with $e_k \in \mathcal{R}$ and $e \geq 0$ are the residual errors, and K is the number of sensor measurements. Where the k-th residual error, e_k, is a distance between the k-th measurement point and a corresponding landmark.

In this work, we divide the sensor measurements into three classes, aligned, misaligned, and unknown. Because these classes correspond to the latent variables, the number of the latent state is three. “aligned” and “misaligned” mean that the sensor measurements have hit the landmarks and correctly match, or do not match, with them. “unknown” means that the sensor measurements have hit obstacles that do not exist on the map. $p(z_{k,1} = 1)$, $p(z_{k,2} = 1)$, and $p(z_{k,3} = 1)$ denote probabilities of the aligned, misaligned, and unknown classes of the k-th measurement, respectively. We first estimate posterior probabilities of the measurement classes, $p(\mathbf{Z}|\mathbf{e})$, and then detect the localization failure based on these probabilities.

A. Measurement class estimation via posterior distributions

The posterior probabilities are denoted as:

$$\begin{aligned} p(\mathbf{Z}|\mathbf{e}) =& \frac{1}{Z} \prod_{i=1}^{K} \prod_{l=1}^{3} p(e_i|\mathbf{z}_i)^{z_{i,l}} \\ & \prod_{i=1}^{K} \prod_{j=1\backslash i}^{K} \prod_{l=1}^{3} \prod_{m=1}^{3} \psi_{i,j}(\mathbf{z}_i, \mathbf{z}_j)^{z_{i,l} z_{j,m}}, \end{aligned} \tag{2}$$

where Z is a normalization constant, $p(e_i|\mathbf{z}_i)$ represent likelihood distributions, $\psi_{i,j}(\mathbf{z}_i, \mathbf{z}_j)$ is a transition probability matrix between a maximal clique composed of $\mathbf{z}_i$ and $\mathbf{z}_j$, and $j = 1\backslash i$ are integral numbers with the exception of i. The likelihood distributions are described in Section III-B. Estimating the probabilities is not practical because the number of states of the probabilities is 3^K. Instead, we individually estimate marginal posterior probabilities of $\mathbf{z}_k$ as:

$$p(\mathbf{z}_k|\mathbf{e}) = \sum_{\mathbf{z}_1} \cdots \sum_{\mathbf{z}_{k-1}} \sum_{\mathbf{z}_{k+1}} \cdots \sum_{\mathbf{z}_K} p(\mathbf{z}|\mathbf{e}). \tag{3}$$

If the latent variables are connected by only its vicinity variables, i.e., with no loop connections, then the exact marginal posterior probability can be efficiently calculated as follows:

$$p(\mathbf{z}_k|\mathbf{e}) = \frac{1}{Z} \mathbf{l}_k \otimes \boldsymbol{\mu}_\alpha(\mathbf{z}_k) \otimes \boldsymbol{\mu}_\beta(\mathbf{z}_k), \tag{4}$$

where $\otimes$ is the Hadamard product operator, $\mathbf{l}_k$ is a likelihood vector, and $\boldsymbol{\mu}_\alpha$ and $\boldsymbol{\mu}_\beta$ are forward and backward messages [31]. The messages can be recursively calculated as follows:

$$\begin{aligned} \boldsymbol{\mu}_\alpha(\mathbf{z}_k) =& \psi_{k-1,k}(\mathbf{z}_{k-1}, \mathbf{z}_k) \\ & \mathbf{l}_{k-1} \otimes \psi_{k-2,k-1}(\mathbf{z}_{k-2}, \mathbf{z}_{k-1}) \cdots, \\ =& \psi_{k-1,k}(\mathbf{z}_{k-1}, \mathbf{z}_k) \boldsymbol{\mu}_\alpha(\mathbf{z}_{k-1}), \end{aligned} \tag{5}$$

$$\begin{aligned} \boldsymbol{\mu}_\beta(\mathbf{z}_k) =& \psi_{k+1,k}(\mathbf{z}_{k+1}, \mathbf{z}_k) \\ & \mathbf{l}_{k+1} \otimes \psi_{k+2,k+1}(\mathbf{z}_{k+2}, \mathbf{z}_{k+1}) \cdots, \\ =& \psi_{k+1,k}(\mathbf{z}_{k+1}, \mathbf{z}_k) \boldsymbol{\mu}_\beta(\mathbf{z}_{k+1}), \end{aligned} \tag{6}$$

and the first values are calculated as:

$$\boldsymbol{\mu}_\alpha(\mathbf{z}_2) = \psi_{1,2}(\mathbf{z}_1, \mathbf{z}_2)(\mathbf{l}_1 \otimes \mathbf{1}), \tag{7}$$

$$\boldsymbol{\mu}_\beta(\mathbf{z}_{K-1}) = \psi_{K,K-1}(\mathbf{z}_K, \mathbf{z}_{K-1})(\mathbf{l}_K \otimes \mathbf{1}). \tag{8}$$

The proposed graphical model, however, has loop connections and, thus, the exact marginal posterior probabilities cannot be calculated using the aforementioned equations. Approximate marginal posterior probabilities of the latent variables can be estimated using loopy belief propagation [31]. First, every latent variable receives messages from its connected variables to initialize the marginal posterior probabilities:

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{e}) =& \frac{1}{Z} \mathbf{l}_k \otimes \boldsymbol{\mu}'_{1\to k}(\mathbf{z}_k) \otimes \cdots \otimes \boldsymbol{\mu}'_{k-1\to k}(\mathbf{z}_k) \\ & \otimes \boldsymbol{\mu}'_{k+1\to k}(\mathbf{z}_k) \otimes \cdots \otimes \boldsymbol{\mu}'_{K\to k}(\mathbf{z}_k), \end{aligned} \tag{9}$$

where $\boldsymbol{\mu}'_{i\to k}(\mathbf{z}_k)$ is the initial message sent from i-th to k-th latent variables and is denoted as follows:

$$\boldsymbol{\mu}'_{i\to k}(\mathbf{z}_k) = \psi_{i,k}(\mathbf{z}_i, \mathbf{z}_k)(\mathbf{l}_i \otimes \mathbf{1}). \tag{10}$$

After every latent variable receives the messages, the variables send further messages using an arbitrary passing strategy. The marginal posterior probabilities are updated until convergence:

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{e}) \leftarrow & \frac{1}{Z} p(\mathbf{z}_k|\mathbf{e}) \otimes \psi_{i,k}(\mathbf{z}_i, \mathbf{z}_k) p(\mathbf{z}_i|\mathbf{e}), \\ =& \frac{1}{Z} p(\mathbf{z}_k|\mathbf{e}) \otimes \boldsymbol{\mu}_{i\to k}(\mathbf{z}_k). \end{aligned} \tag{11}$$

B. Likelihood distributions of residual errors

The bottom of Fig. 1 shows the histograms of residual errors for the aligned, misaligned, and unknown classes. These histograms were built using a simulator and the ground truth was available. The residual errors of the aligned, misaligned, and unknown classes are in accordance with normal (red), exponential (blue), and uniform (green) distributions, respectively. We model the likelihood distributions as:

$$\prod_{l=1}^{3} p(e_k|\mathbf{z}_k)^{z_{k,l}} = \prod_{l=1}^{3} p(e_k|z_{k,l} = 1, \theta_l)^{z_{k,l}}, \tag{12}$$

$$p(e_k|z_{k,1} = 1, \theta_1) = 2\mathcal{N}(e_k; 0, \sigma^2), \tag{13}$$

$$p(e_k|z_{k,2} = 1, \theta_2) = \frac{\lambda \exp\left(-\lambda e_k\right)}{1 - \exp\left(-\lambda e_{\max}\right)}, \tag{14}$$

$$p(e_k|z_{k,3} = 1, \theta_3) = \mathrm{unif}(0, e_{\max}), \tag{15}$$

where $e_{\max}$ is the maximum residual error, θ_l is the hyperparameter of the l-th likelihood distribution, i.e., $\theta_1 = \sigma^2$, $\theta_2 = \lambda$, and $\theta_3 = \emptyset$, $\mathcal{N}(e_i; 0, \sigma^2)$ is the zero-mean Gaussian with a variance of σ^2, and $\mathrm{unif}(0, e_{\max})$ is the uniform distribution within the given region. The factor 2 in (13) stems from the fact that the residual is only defined for positive numbers, which means that N only integrates to 0.5. This also means that the sign of the residual errors can be ignored.

C. Localization failure detection

Equation (11) gives us probabilities of the k-th sensor measurement classes; these do not directly provide a criterion for the detection of localization failures. We need to determine a threshold to detect failures based on the probabilities.

In this work, we use a misalignment ratio as the threshold for detecting a failure. The misalignment ratio is defined as:

$$\mathrm{MIS} = \frac{\sum_{i=1}^{K} z_{k,2}}{K - \sum_{k=1}^{K} z_{k,3}}, \tag{16}$$

where $z_{k,2} = 1$ and $z_{k,3} = 1$ denote that the k-th sensor measurement class is misaligned and unknown and $K - \sum_{k=1}^{K} z_{k,3}$

denotes that the unknown classes are ignored to calculate the misalignment ratio. We define the localization failure whereby the MIS is larger than the threshold, $\mathrm{MIS}_{\mathrm{th}}$. Thus, the localization failure probability, p_{failure}, is denoted as:

$$p_{\mathrm{failure}} = \sum_{\mathbf{Z} \in C} p(\mathbf{Z}|\mathbf{e}), \tag{17}$$

where $\mathbf{Z} \in C$ is a set of states that satisfy the condition, $\mathrm{MIS} \geq \mathrm{MIS}_{\mathrm{th}}$. It is, however, not practical to compute the probability because the number of the states is 3^K. We use a sampling-based method to approximately compute it:

$$\hat{\mathbf{z}}_k \sim p(\mathbf{z}_k|\mathbf{e}), \tag{18}$$

$$p_{\mathrm{failure}} \simeq \frac{1}{N} \sum_{n=1}^{N} \mathbb{1} \left(\frac{\sum_{i=1}^{K} \hat{z}_{k,2}}{K - \sum_{k=1}^{K} \hat{z}_{k,3}} \geq \mathrm{MIS}_{\mathrm{th}} \right), \tag{19}$$

where N is the number of sampling processes and $\mathbb{1}(\cdot)$ is an indicator function set at 1 when the condition within the bracket is true, and 0 otherwise.

IV. Experimental conditions

During the experiments, we focused on the 2D LiDAR-based localization problem and used a 2D LiDAR simulator[1]. Thanks to the use of a simulator, we could use the ground truth data and were hence able to exactly evaluate performance. We created datasets composed of success and failure localization samples and validated the classification accuracy of these samples. We did not focus on the continuous time series localization problem because setting constant conditions in each sample is difficult. Instead, we focused on the classification of localization estimation in a certain moment. Examples of the success and failure localization estimation samples are shown in Fig. 4. The accompanying video shows the demonstrations of the proposed method of the continuous localization problem utilizing a noisy odometry simulation.

The accompanying video also shows a demonstration with the real robot. The video only shows the qualitative results, but it serves to assist in the understanding of the performance.

A. 2D LiDAR simulator

We began by building 2D occupancy grid maps (OGMs) for real environments using the 2D LiDAR SLAM [4]. To reproduce the dynamic environment in a simulator, we simulated unknown obstacles that randomly walk on free spaces of the OGM and randomly remove landmarks. The measurements of the 2D LiDAR were simulated on the OGM. The Gaussian noise with $(0.03\ \mathrm{m})^2$ variance was added to the measurement ranges. We referred to the Hokuyo 2D LiDAR (TOP-URG) and the following detailed specifications of the 2D LiDAR: the measurement range was 30 m, the measurement angle was 270 deg, and the measurement angle resolution was 0.25 deg.

TABLE I
Estimation and residual errors of the datasets.

		Ave.	Std.	Min.	Max.
Success	Trans. error	0.08 m	0.03 m	0 m	0.14 m
	Head. error	0.25 deg	0.14 deg	0 deg	0.50 deg
	Resid. error	0.05 m	0.05 m	0 m	4.82 m
Failure	Trans. error	0.23 m	0.06 m	0 m	0.59 m
	Head. error	0.77 deg	0.70 deg	0 deg	3.74 deg
	Resid. error	0.10 m	0.10 m	0 m	5.22 m

TABLE II
Ratios related to the measurements of the datasets.

	Ave.	Std.	Min.	Max.
Valid measurement	94.35 %	7.33 %	40.43 %	100 %
Valid landmark	95.65 %	5.71 %	47.27 %	100 %
Landmark measurement	54.18 %	18.03 %	0.74 %	100 %
Unknown measurement	40.17 %	18.21 %	0 %	98.89 %

B. Datasets

1) Success and failure localization samples: We defined a failure in the localization to be when the translation or heading direction errors, $\sqrt{\Delta x^2 + \Delta y^2}$ and $|\Delta\theta|$, exceed thresholds. These thresholds were set to 0.2 m and 2 deg. The failure localization samples were created by adding Gaussian noise to the ground truth. An example of the failure sample is shown in the bottom of Fig. 4. Slight mismatches can be seen.

The ground truth data are not used for the success localization samples because they do not have any estimation errors and this is not realistic. We add small noise to the ground truth while the translation and heading direction errors do not exceed the thresholds. Hence, the slightly disturbed ground truth is used as the success samples. Its example is also seen at the top of Fig. 4.

2) Details of datasets: We used five indoor environments including a larger car garage, two lobbies, and two office floors to create the datasets. The LiDAR can measure enough landmarks at all of the areas in the environments as shown in Fig. 1 (the Fig. 1 shows the car garage). Approximate size of the environments are shown in Table III.

Table I shows the estimation and residual errors of the success and failure datasets. The values related to the residual errors were calculated using only measurements that hit the landmarks. The success and failure datasets include 2394 samples, respectively. As can be seen from the table, the estimation and residual errors of the failure datasets are not large. Hence, slight mismatches should be recognized to accurately classify the localization samples.

Table II shows ratios related to the sensor measurements. “Valid measurement” is the ratio of whether the measurements hit or do not hit objects. “Valid landmark” is the ratio of the landmarks that are located in the measurement range. “Landmark measurement” and “Unknown measurement” are ratios of the measurements that hit the landmarks and unknown obstacles. As can be seen from the table, the localization can be suggested to have performed well because the valid landmarks are enough and almost half of the measurements hit the landmarks. Meanwhile, almost half of the measurements hit unknown obstacles. Hence, robustness against environmental changes is also necessary for a better overall performance.

[1] https://github.com/NaokiAkai/AutoNavi. This repository includes map data used for creating the datasets.

V. Experiments

A. Parameters for the proposed method

1) Transition probabilities: We use the homogeneous transition probability matrix, denoted as:

$$\psi_{i,j}(\mathbf{z}_i, \mathbf{z}_j) = \begin{pmatrix} 4/5 & 0 & 1/5 \\ 0 & 4/5 & 1/5 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}. \tag{20}$$

The first, second, and third rows are the transition probabilities from the aligned, misaligned, and unknown classes, respectively. We assumed that the measurements, with the exception of measurements hitting unknown obstacles, belong to the aligned (misaligned) classes if the i-th measurement belongs to the aligned (misaligned) class as these classes do not simultaneously appear. Hence, these transition probabilities are not uniform. By contrast, we assumed that the unknown class does not give us information to predict the other measurement classes, because the measurements that hit unknown obstacles cannot be usually expected. Hence, the transition probability is set to uniform and the matrix is not symmetric.

2) Likelihood distributions: We set the parameters relating to the likelihood distributions as $\sigma^2 = (0.075 \text{ m})^2$, $\lambda = 10.1$, and $e_{\max} = 0.6$ m. These parameters were tuned using corresponding datasets that were not used for the validation.

3) Misalignment ratio threshold and failure detection: We set the misalignment ratio threshold, $\mathrm{MIS}_{\mathrm{th}}$, to 10 %. In following the comparison, we further investigated the effect caused by the threshold. The localization results were categorized as a failure if the failure probability, p_{failure}, was larger than 0.5. The number of the sampling process, N, was set to 1000.

4) Computation process: The voxel grid filter with resolution 0.1 m was applied to the sensor measurements to ensure spatial uniformity and the residual errors were computed. Then, the MRFs with the FCLVs took the residual errors and performed the iterative computation process for estimating the marginal posterior probabilities, as shown in equation (11), until convergence occurred. Summation of the variation of the probabilities was checked and the process was regarded as converged if the summation within 100 steps was less than 10^{-9}. During implementation, the random update strategy was used as the loopy belief propagation.

B. Comparison methods

We compare the proposed method with two model, and four ML, based methods. The model-based methods did not require training data and the proposed method was also the model-based method. The MRFs without the FCLVs (MRFs w/o FCLVs) and root mean square-based classification were used as the model-based methods. The latent variables of the MRFs w/o FCLVs were only connected to its neighbouring variables. Logistic regression, support vector machine (SVM), AdaBoost, and CNN were used as the ML-based methods.

1) Input data and output label: All the methods, with the exception of logistic regression and SVM, used a vector of residual errors, $\mathbf{e}$, as the input data. The logistic regression and SVM used handcrafted features as the input data (see Section V-B5 for more details). We used a binary label $l \in \{0, 1\}$ and provided $l = 1$ and $l = 0$ to the success and failure samples.

2) Learning and validation: The ML-based methods were trained in a supervised manner. In comparison, datasets created from four differing environments were used for the training and the single remaining dataset was used for the validation.

3) MRFs w/o FCLVs: We calculated the marginal posterior probabilities using the equation (4). Then, the localization failure probability was calculated using the equation (19) and the same misalignment ratio threshold as used in the proposed method. The localization samples were categorized as a failure if the failure probability, p_{failure}, was larger than 0.5.

4) Root mean square: The root mean square is denoted as:

$$\mathrm{RMS} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} e_k^2}. \tag{21}$$

We computed averages of the RMS for both success and the failure localization datasets. The resultant averages were used as the thresholds for distinguishing the localization samples.

5) Logistic regression and SVM: We created features from the residual errors for the processes of logistic regression and SVM. Sum, average, and standard deviation of the errors and vicinity error difference were used for these features.

6) AdaBoost: The decision tree classifier was used for the learning model of weak classifiers. The number of weak classifiers was set to 1000.

7) CNN: The CNN for comparing image similarity are proposed in [32]. Based on the CNN, we previously proposed the CNN to distinguish whether the localization has failed or not in [24], [25]. We use the CNN in this work.

The architecture of the CNN was composed of four convolution and max pooling layers and two fully connected layers. The CNN received the residual errors and output a continuous value from zero to one. The localization samples were categorized as failure when the output was less than 0.5.

C. Comparison results of classification performance

Table III shows classification results of the success and failure localization samples. The values shown in the table were computed using the numbers of true positives, false positives, true negatives, and false negatives, respectively. Here, the success and failure samples are used as the positive and negative samples. The A, B, C, D, and E are names of the environment and the total means all datasets. The same number of success and failure samples were used for the validation, i.e., the chance rate was 50 %. It should be noted that the recall and F-measure values from the MRFs w/o FCLVs were not defined because all samples were classified as being a failure and the denominator values were zero.

In the environments A, C, D, and E, the proposed method was the most accurate. In the environment B, the CNN were the most accurate. For the total data, the proposed method was the most accurate and its accuracy exceeded 95 %. The AdaBoost had a good performance, as presented in [19], but its performance was marginally lower than that of the proposed method and CNN. The root mean square-based classification

TABLE III
CLASSIFICATION RESULTS.

		Proposed Method	MRFs w/o FCLVs	Root mean square	Logistic regression	SVM	AdaBoost	CNN
	Accuracy	**97.76 %**	50.00 %	82.36 %	60.33 %	63.64 %	93.86 %	96.49 %
A	Precision	**98.64 %**	0 %	95.13 %	77.58 %	66.67 %	95.71 %	97.86 %
Car garage	Recall	**96.93 %**	-	75.78 %	57.68 %	62.87 %	92.29 %	95.26 %
30 m × 35 m	Specificity	**98.61 %**	50.00 %	93.46 %	65.77 %	64.52 %	95.55 %	97.80 %
	F-measure	**97.78 %**	-	84.36 %	66.17 %	64.71 %	93.97 %	96.54 %
	Accuracy	95.51 %	50.00 %	75.93 %	62.69 %	64.22 %	91.90 %	**97.05 %**
B	Precision	93.00 %	0 %	65.86 %	60.39 %	62.80 %	89.72 %	**96.50 %**
Lobby 1	Recall	**97.93 %**	-	82.47 %	63.30 %	64.64 %	93.82 %	97.5 %
40 m × 20 m	Specificity	93.33 %	50.00 %	71.58 %	62.13 %	63.83 %	90.15 %	**96.54 %**
	F-measure	95.40 %	-	73.24 %	61.81 %	63.70 %	91.72 %	**97.03 %**
	Accuracy	**92.79 %**	50.00 %	74.45 %	54.91 %	59.06 %	87.34 %	92.36 %
C	Precision	**98.25 %**	0 %	92.58 %	63.54 %	65.28 %	94.32 %	94.54 %
Office floor 1	Recall	88.58 %	-	67.95 %	54.19 %	58.06 %	87.76 %	**90.59 %**
50 m × 50 m	Specificity	**98.04 %**	50.00 %	88.36 %	55.94 %	60.35 %	93.40 %	94.29 %
	F-measure	**93.17 %**	-	78.37 %	58.49 %	61.46 %	88.16 %	92.52 %
	Accuracy	**97.02 %**	49.90 %	77.52 %	58.73 %	62.73 %	92.40 %	96.61 %
D	Precision	98.35 %	0 %	74.54 %	52.36 %	58.52 %	97.13 %	**98.97 %**
Lobby 2	Recall	**95.80 %**	-	79.26 %	60.00 %	63.90 %	88.74 %	94.51 %
50 m × 20 m	Specificity	98.31 %	49.95 %	75.97 %	57.74 %	61.74 %	96.83 %	**98.92 %**
	F-measure	**97.06 %**	-	76.83 %	55.92 %	61.09 %	92.75 %	96.70 %
	Accuracy	**92.90 %**	50.00 %	76.19 %	57.41 %	58.14 %	87.58 %	91.03 %
E	Precision	**90.40 %**	0 %	70.14 %	38.41 %	38.83 %	87.27 %	87.27 %
Office floor 2	Recall	**95.16 %**	-	79.62 %	61.95 %	63.27 %	87.82 %	94.37 %
40 m × 50 m	Specificity	**90.85 %**	50.00 %	73.32 %	55.37 %	55.87 %	87.34 %	88.16 %
	F-measure	**92.72 %**	-	74.58 %	47.42 %	48.12 %	87.54 %	90.67 %
	Accuracy	**95.28 %**	49.98 %	77.38 %	58.83 %	61.59 %	90.69 %	94.70 %
	Precision	**95.78 %**	0 %	79.87 %	58.65 %	58.44 %	92.90 %	95.01 %
Total	Recall	**94.83 %**	-	76.08 %	58.87 %	63.37 %	88.96 %	94.37 %
	Specificity	**95.74 %**	50.00 %	78.81 %	58.80 %	60.90 %	92.57 %	95.04 %
	F-measure	**95.30 %**	-	77.93 %	58.76 %	60.34 %	90.89 %	94.67 %

worked well in some environments, but its total accuracy was less than 80 %. The MRFs w/o FCLVs, logistic regression, and SVM did not have a good performance.

It is not easy to argue through the results that the performance of the proposed method is statistically superior to that of the CNN because the classification accuracies of the two methods are almost the same. However, the important result is that the proposed method with the explainable models yielded almost the same performance as the CNN. We consider that the explainable models are more useful than CNN-based models if their classification performance is almost the same.

D. Performance investigation of the proposed method

1) Computational complexity: We measured computational complexity as described in Section V-A4. An Intel(R) Xeon(R) E5-1650 v3@3.50 GHz was used and the average/standard deviations of the iteration number and computational time were 318.91/415.90 and 14.25/4.32 msec. The computational time of the proposed method is longer than that of other methods, as only the proposed method employed an iteration process. However, the proposed method can work in real time because the measurement frequency of the sensor is 40 Hz.

2) Effect of the hyperparameters: Figure 3 shows the recognition accuracy from the proposed method when applied to different hyperparameters as shown in the equations (13) and (14). σ was changed from 0.05 to 0.1 every 0.005 and λ was changed from 7 to 12 every 0.5. The highest accuracy was 95.24 % and the parameters were $\sigma = 0.075$ and $\lambda = 10.0$. The classification accuracy did not produce a drastic decrease, even though the parameters were slightly changed.

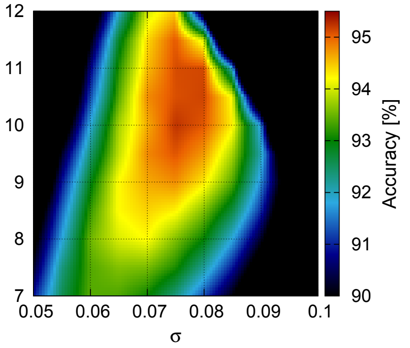

Fig. 3. The recognition accuracy from the proposed method with different hyperparameters for the pre-determined likelihood distributions.

3) Measurement class recognition accuracy: Table IV shows the averages and standard deviations of the measurement class recognition accuracy using the proposed method and the MRFs w/o FCLVs. The class recognition accuracy using the proposed method exceeded 90 % for the success, failure, and both datasets. The recognition accuracy from the MRFs w/o FCLVs for the failure dataset was quite bad.

Figures 4 and 5 show examples of the measurement class recognition results using the proposed method and the MRFs w/o FCLVs. The proposed method accurately recognized the measurement classes. As can be seen in the bottom right of Fig. 4, a slight mismatch could be exactly identified even though the partial measurements were overlapped with the landmarks. The recognition results of the MRFs w/o FCLVs, when the localization process was a success, were good but were a bit noisy. Hence, almost all successful localization re-

TABLE IV

AVERAGES AND STANDARD DEVIATIONS OF MEASUREMENT CLASS RECOGNITION ACCURACY USING THE PROPOSED METHOD AND MRFS W/O FCLVS.

	Proposed method			MRFs w/o FCLVs		
Dataset	Success	Failure	Both	Success	Failure	Both
A	97.22 (6.25) %	93.71 (11.71) %	95.37 (9.39) %	93.69 (4.46) %	59.31 (9.57) %	76.50 (7.47) %
B	93.88 (11.29) %	91.81 (12.13) %	92.48 (11.72) %	91.92 (5.05) %	55.36 (11.73) %	73.64 (9.03) %
C	96.05 (9.53) %	85.34 (25.65) %	90.70 (19.35) %	92.80 (5.84) %	42.56 (10.95) %	67.68 (8.77) %
D	97.07 (6.59) %	91.98 (13.95) %	94.53 (10.91) %	93.49 (4.65) %	56.14 (12.41) %	74.81 (9.37) %
E	91.17 (6.73) %	88.14 (16.47) %	89.66 (17.19) %	90.11 (6.73) %	51.32 (12.51) %	70.72 (10.05) %
Total	95.07 (11.32) %	90.12 (16.94) %	92.59 (14.41) %	92.43 (5.55) %	53.11 (12.83) %	72.77 (9.88) %

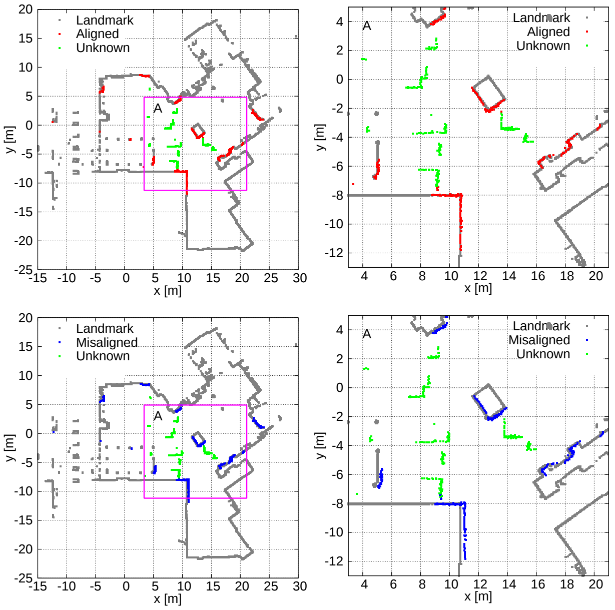

Fig. 4. Measurement class recognition results in the success (top) and failure (bottom) cases by the proposed method. The right hand figures are enlarged, and figures in area A are shown to the left.

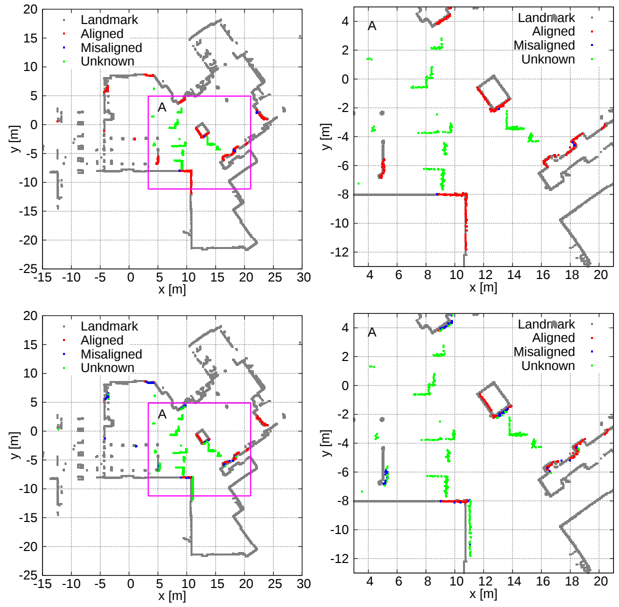

Fig. 5. Measurement class recognition results in the success (top) and failure (bottom) cases by the MRFs without the FCLVs.

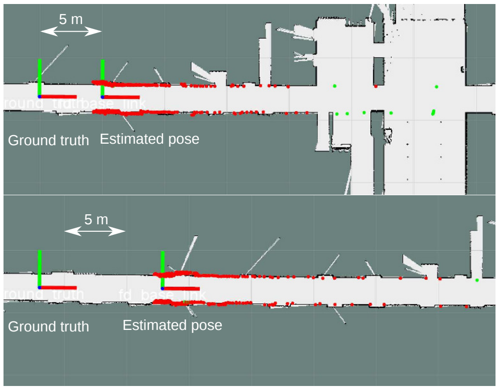

Fig. 6. Misclassification results using the proposed method. These estimates were categorized as a success even though the estimation errors were large.

sults were classified as a failure because the misalignment ratio threshold was not large (10 %). The recognition accuracy was quite bad when the localization had failed because almost all of the measurements that did not overlap with the landmarks were categorized as the unknown class.

4) Effect of the misalignment ratio thresholds: Table V shows the classification results from the proposed method and MRFs w/o FCLVs with different misalignment ratios. The MRFs w/o FCLVs worked well if the ratio was set to 50 %. By contrast, the proposed method worked well with all of the ratios. Because the proposed method was able to accurately recognize the measurement classes as shown in Fig. 4, the threshold did not drastically influence the overall performance. The Figs. 4 and 5 and Table V show that the FCLVs are effective in improving the class recognition accuracy.

E. Limitations

The proposed method has some limitations. Figure 6 shows examples of such limitations. The top of Fig. 6 shows a localization result with large estimation errors, of around 5 m. The misaligned measurements can be seen on the right side (green points), but these measurements were categorized as the unknown class because the residual errors were not in accordance with the expected likelihood distributions. Consequently, the proposed method cannot distinguish this estimation as the failure even though some mismatches are seen. The bottom of Fig. 6 also shows a localization result with large estimation errors, but here all the measurements overlap with the landmarks. In such a case, the residual errors are according to the normal distribution and the proposed method cannot detect such failures.

The accompanying video shows a demonstration with large

TABLE V
CLASSIFICATION RESULTS BY THE PROPOSED METHOD AND MRFS W/O FCLVS WITH DIFFERENT MISALIGNMENT RATIO THRESHOLDS.

	Proposed method				MRFs w/o FCLVs			
Threshold (Misalignment ratio)	30 %	50 %	70 %	90 %	30 %	50 %	70 %	90 %
Accuracy	95.28 %	95.28 %	95.26 %	95.22 %	50.00 %	93.42 %	51.82 %	50.00 %
Precision	95.82 %	95.82 %	95.82 %	95.74 %	0.13 %	98.37 %	100 %	100 %
Recall	94.79 %	94.79 %	94.75 %	94.75 %	50.00 %	89.51 %	50.93 %	50.00 %
Specificity	95.78 %	95.78 %	95.77 %	95.69 %	50.00 %	98.19 %	100 %	-
F-measure	95.30 %	95.31 %	95.29 %	95.24 %	0.25 %	93.73 %	67.48 %	66.67 %

estimation errors and whereby the proposed method distinguishes these localization results with large estimation errors as a failure. However, It is difficult to guarantee positive performance of the proposed method in large estimation error cases because the residual errors may not obey the predetermined likelihood distributions, as shown in the top of Fig. 6. Hence, the proposed method is suitable for immediately detecting failure of the position tracking problem using filtering-based localization, for example Kalman and particle filters [1], because here the estimation errors slightly increase.

VI. CONCLUSION

This paper has proposed a misalignment recognition method using MRFs with FCLVs for the detection of localization failure. In the localization problem, we generally assume that sensor measurements are independent for simply modeling the measurements. This assumption, however, causes a problem in which entire relations of the measurements are ignored. Consequently, it is hard to recognize misalignment because of localization failure. The full connection of the MRFs allowed us to consider the entire relation of the measurements and we achieved exact misalignment recognition even though partial sensor measurements were overlapped with landmarks. This paper has also presented a solution for detecting localization failure based on the misalignment recognition.

The proposed method was compared to six existing methods, including CNN, by using datasets composed of success and failure localization samples. In the comparative experiments, we investigated classification accuracy in all of the methods. The experimental results showed that classification accuracy using the proposed method exceeded 95 % and outperformed the other examined methods. Future work includes integrating the proposed method with simultaneous localization and reliability estimation method that we previously proposed in [24], [25].

ACKNOWLEDGMENT

This work was supported by the Center of Innovation Program funded by the JST and KAKENHI 40786092.

REFERENCES

[1] S. Thrun et al. *Probabilistic Robotics*. The MIT Press, 2005.

[2] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proc. of the IEEE ICRA*, volume 2, pages 674–680, 1989.

[3] B. Yamauchi et al. Place recognition in dynamic environments. *J. of Robotic Systems*, 14:107–120, 1997.

[4] G. Grisetti et al. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. on Robotics*, 23(1):34–46, 2007.

[5] C.-C. Wang et al. Simultaneous localization, mapping and moving object tracking. *The Int. J. of Robotics Research*, 26(9):889–916, 2007.

[6] J. Gutmann et al. An experimental comparison of localization methods continued. In *Proc. IEEE/RSJ IROS*, pages 454–459, 2002.

[7] P. J. Besl et al. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[8] P. Biber et al. The normal distributions transform: A new approach to laser scan matching. In *Proc. of the IEEE/RSJ IROS*, pages 2743–2748, 2003.

[9] A. Birk. A quantitative assessment of structural errors in grid maps. *Autonomous Robots*, 28:187–196, 2010.

[10] S. Schwertfeger et al. Evaluation of map quality by matching and scoring high-level, topological map structures. In *Proc. of the IEEE ICRA*, pages 2221–2226, 2013.

[11] S. Rusinkiewicz et al. Efficient variants of the ICP algorithm. In *Proc. Int. Conf. on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[12] A. Makadia et al. Fully automatic registration of 3D point clouds. In *Proc. of the IEEE CVPR*, volume 1, pages 1297–1304, 2006.

[13] L. Silva et al. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(5):762–776, 2005.

[14] P. Sundvall et al. Fault detection for mobile robots using redundant positioning systems. In *Proc. of the IEEE ICRA*, pages 3781–3786, 2006.

[15] J. P. Mendoza et al. Mobile robot fault detection based on redundant information statistics. In *Proc. of the IEEE/RSJ IROS*, 2012.

[16] Z. Alsayed et al. 2D SLAM correction prediction in large scale urban environments. In *Proc. of the IEEE ICRA*, pages 5167–5714, 2018.

[17] S. Nobili et al. Predicting alignment risk to prevent localization failure. In *Proc. of the IEEE ICRA*, pages 1003–1010, 2018.

[18] L. T. Hsu. GNSS mulitpath detection using a machine learning approach. In *Proc. of the IEEE ITSC*, pages 1414–1419, 2017.

[19] H. Almqvist et al. Learning to detect misaligned point clouds. *J. of Field Robotics*, 35:662–677, 2018.

[20] Y. Freund et al. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, 55(1):119–139, 1997.

[21] A. Kendall et al. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proc. of the IEEE ICCV*, pages 2938–2946, 2015.

[22] W. Yuan et al. Iterative transformer network for 3D point cloud. *arXiv:1811.11209*, 2018.

[23] N. Akai et al. Mobile robot localization considering class of sensor observations. In *Proc. of the IEEE/RSJ IROS*, pages 3159–3166, 2018.

[24] N. Akai et al. Reliability estimation of vehicle localization result. In *Proc. of the IEEE IV*, pages 740–747, 2018.

[25] N. Akai et al. Simultaneous pose and reliability estimation using convolutional neural network and Rao-Blackwellized particle filter. *Advanced Robotics*, 32(17):930–944, 2018.

[26] F. T. Ramos et al. CRF-matching: Conditional random fields for feature-based scan matching. In *RSS*, 2007.

[27] M. Chandran-Ramesh et al. Assessing map quality using conditional random fields. In *FSR*, volume 42, 2007.

[28] J. Stechschulte et al. Hidden Markov random field iterative closest point. *arXiv:1711.05864*, 2017.

[29] D. Wang et al. Accurate mix-norm-based scan matching. In *Proc. of the IEEE/RSJ IROS*, pages 1665–1671, 2018.

[30] N. Akai et al. Toward localization-based automated driving in highly dynamic environments: Comparison and discussion of observation models. In *Proc. of the IEEE ITSC*, pages 2215–2222, 2018.

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

[32] S. Zagoruyko et al. Learning to compare image patches via convolutional neural networks. In *Proc. of the IEEE CVPR*, pages 4353–4361, 2015.