

Article

HMM-Based Dynamic Mapping with Gaussian Random Fields [†]

Hongjun Li ^{1,*}, Miguel Barão ², Luís Rato ² and Shengjun Wen ¹

¹ Zhongyuan-Petersburg Aviation College, Zhongyuan University of Technology, Zhengzhou 450007, China; wsj@zut.edu.cn

² Departamento de Informática, Escola de Ciências e Tecnologia, Universidade de Évora, 7004-516 Évora, Portugal; mjsb@uevora.pt (M.B.); lmr@uevora.pt (L.R.)

* Correspondence: li.hongjun@zut.edu.cn; Tel.: +86-13526706493

[†] This paper is an extended version of our paper published in 24th International Conference on Automation and Computing under the title “Mapping Dynamic Environments Using Markov Random Field Models”.

Abstract: This paper focuses on the mapping problem for mobile robots in dynamic environments where the state of every point in space may change, over time, between free or occupied. The dynamical behaviour of a single point is modelled by a Markov chain, which has to be learned from the data collected by the robot. Spatial correlation is based on Gaussian random fields (GRFs), which correlate the Markov chain parameters according to their physical distance. Using this strategy, one point can be learned from its surroundings, and unobserved space can also be learned from nearby observed space. The map is a field of Markov matrices that describe not only the occupancy probabilities (the stationary distribution) as well as the dynamics in every point. The estimation of transition probabilities of the whole space is factorised into two steps: The parameter estimation for training points and the parameter prediction for test points. The parameter estimation in the first step is solved by the expectation maximisation (EM) algorithm. Based on the estimated parameters of training points, the parameters of test points are obtained by the predictive equation in Gaussian processes with noise-free observations. Finally, this method is validated in experimental environments.

Keywords: dynamic environments; Markov chain; Gaussian random fields; expectation maximisation



Citation: Li, H.; Barão, M.; Rato, L.; Wen, S. HMM-Based Dynamic Mapping with Gaussian Random Fields. *Electronics* **2022**, *11*, 722. <https://doi.org/10.3390/electronics11050722>

Academic Editor: Enzo Pasquale Scilingo

Received: 14 January 2022

Accepted: 21 February 2022

Published: 26 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Literature Review

Dynamic environments are particularly important and complex. These environments include static objects and different kinds of dynamic objects. High dynamic objects, such as moving people, change their position quickly. Low dynamic objects, such as doors and pieces of furniture, can appear and disappear from particular locations however those events are comparatively rare. Autonomous robots should be able to know if objects are static or dynamic to help in path planning.

In earlier research, the environments were assumed to be static. The classical method for static environments is occupancy grid mapping [1–3] where maps are divided into a grid and the states of different grid cells are assumed to be independent. In dynamic environments, one popular strategy is to estimate the number of potential targets, their positions, and velocities from sensor data [4–6]. The dynamic object detection needs to identify the objects and their correspondence in different time instants. The other one is to apply Markov chains. The dynamic occupancy grids proposed in [7–13] does not rely on high-level object models. Every grid cell is associated with a Markov chain, where its future occupancy state only depends on the current state. Since the occupancy observations are noisy, the states are not directly observable and the process is modelled instead by hidden Markov models (HMMs) [14] at each point in space. Estimating good parameters in an HMM requires considerable data. If the dependence between different grid cells is taken into account, as is done in [8], maps are built with inconsistencies.

1.2. Research Gap and Motivation

Dynamic object detection requires more powerful sensors [15], such as cameras, while HMM-based methods can be applied with simple distance sensors. Normally, the correlation between parameters in space is not considered in HMM-based methods and inconsistent maps will be produced. In our previous work [16], the inconsistency in discrete space was dealt with by using Markov random fields to regularise the grid. The static mapping methods [17–23], based on Gaussian Random Fields (GRFs), build smooth occupancy grid maps and predict the occupancy of unobserved space in continuous space. This paper is motivated by them and proposes a new dynamic mapping method based on GRF that is able to deal with continuous space instead of a discrete grid. The state change at every point in continuous space is modelled by a Markov chain with two parameters, and the HMM proposed in [8] is extended with normalised emission probabilities. In one time instant, the map is assumed to be static, and an occupancy grid map is built to obtain the normalised emission probabilities. GRFs are applied to consider the correlation between a point and its neighbouring points. Given the occupancy grid maps, the parameters of every point in the whole space can be estimated. In order to reduce the computation complexity, the parameter estimation is divided into two steps. The first step is to estimate the parameters for training points using the EM algorithm [24,25]. The second step is to predict the parameters for test points using the predictive equation of Gaussian processes with noise-free observations.

1.3. Contribution and Paper Organisation

The main contributions of this paper are highlighted below.

- (1) The extension of the HMM with normalised emission probabilities is developed. Instead of observation models, posterior probability can be used directly to estimate HMM parameters, which is convenient in computation in this paper.
- (2) The HMM parameters of observed space can be smoothed. GRFs are applied to consider the dependence between HMM parameters of different points. The noise in measurements can be filtered and consistent dynamic maps are produced.
- (3) The dynamic behaviour of unobserved space can be predicted. Given the spatial correlation between different points, unobserved space can learn HMM parameters from surrounding observed space and their parameters will be similar.

This paper is organised as follows. The related work is summarised in Section 2. The HMMs with normalised emission probabilities are described in Section 3. The GRF-based methods with known poses and pose uncertainty are proposed in Sections 4 and 5, respectively. The proposed method is validated in experiments in Section 6.

2. Related Work

In dynamic environments, one point in space may have different states in different time instants, and Markov chains can be applied to model the dynamic behaviour. In [8], the map is divided into grid cells, and every grid cell has two possible states: occupied and free. A Markov chain with two parameters is applied to individually modelled every grid cell. One parameter represents the transition probabilities of state from free to occupied. The other one represents the transition probabilities of state from occupied to free. The states can not be observed certainly and an HMM can be applied. In order to deal with incorrect observations, the underlying possible states are extended and consist of seven components: “true”, “false”, “unknown”, “dynamic”, “falsely false”, “falsely true”, and “falsely true/false” in [13]. Based on the parameters of HMMs, dynamic objects can be classified [9]. The dynamic maps based on HMMs are used to do lifelong localisation tasks [10] and simultaneous localisation and mapping [26,27]. In [11], the dynamic behaviour is modelled as an Input-output Hidden Markov Model (IOHMM) [7,28], where the observations of the neighbouring cells in the previous time step are considered, to take the spatial correlation into account. The input of an IOHMM is the observations of neighbouring cells in the previous time step. In [12], the Explicit-state-Duration Hidden

Markov Model (EDHMM) is applied to deal with the Markov chain with variable duration and differentiate the dynamic cells from the static environment.

Gaussian processes, also known as GRFs, can be applied to deal with inconsistency in maps. The advantage is that maps with any resolutions could be built. The Gaussian Process Occupancy Map (GPOM) [17] is an occupancy representation of static environments in continuous space. With the increasing number of training data, the computational complexity of Gaussian processes will also increase. For large-scale environments, training data can be divided into many clusters and a Gaussian process is applied to each subset [18]. Similarly, local Gaussian processes are used to ensure continuity by overlapping clusters [19]. Gaussian processes and Bayesian Committee Machines are applied in [20] to recursively update occupancy maps and surface meshes. The multi-support kernel proposed in [21] enables traditional covariance functions to accept two-dimensional regions, reduces the size of covariance matrices, and accelerates Gaussian process inference and learning. A nested Bayesian committee machine is proposed to learn online 3D occupancy maps using Gaussian processes [22]. Online continuous mapping is proposed to build a map as the zero level set of a Gaussian process implicit surface [23].

3. HMMs for Dynamic Environments

3.1. HMMs

The main difference between static and dynamic environments is the existence of unpredictable dynamic objects. In dynamic environments, one point c in space may be occupied or free in different time instants. Here, its next state m_c^{t+1} is assumed to only depend on the current one m_c^t and a Markov chain is applied to model the dynamic behaviour. The occupied and free states are denoted by s_1 and s_2 , respectively. The Markov chain is shown in Figure 1, where the probability transition matrix for m_c is denoted by $A_c = \{a_{ij}^c\}$ and assumed to be time-invariant.

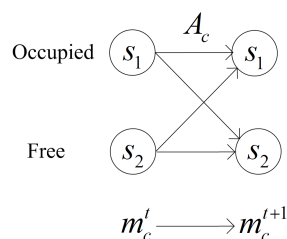


Figure 1. Markov chain for one point.

Defining a grid cell whose central point is c , this point is measured once if one measurement z passes by this grid cell. The Markov chain is a discrete model in time. Between two time instants, the state is assumed to be constant and may be measured multiple times. The measurements for the current state m_c^t are denoted by z_i^t with the same superscript t and different subscripts, and the measurement sequence is denoted by $y_t = (z_1^t, z_2^t, \dots)$. Due to sensor noise, the states cannot be observed certainly, and an HMM can be applied. The graphical model of an HMM is shown in Figure 2, where ζ is the number of the measurement sequences. The corresponding emission probabilities are $p(y_t | m_c^t)$. Since robots always move in space, the observation sequences for the states in different time instants may be different, and the emission probabilities are also different. However, they are not unknown parameters. Assuming independent observations, the emission probabilities can be derived by:

$$p(y_t | m_c^t) = \prod_i p(z_i^t | m_c^t), \tag{1}$$

where $p(z_i^t | m_c^t)$ can be directly given [8].

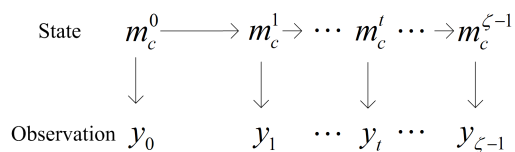


Figure 2. HMM for one point.

The probability of staying in state s_i is a_{ii}^c , and the probability of changing state s_i to the others is $1 - a_{ii}^c$. The probability of staying in state s_i for d time steps from time step t is:

$$p(s_i, d) = (1 - a_{ii}^c)(a_{ii}^c)^{d-1}p(m_c^t = s_i), \tag{2}$$

where $p(m_c^t = s_i)$ is the probability that the state at time t is s_i . The overall expected duration [29] is defined by:

$$\begin{aligned} E(d) &= \sum_{s_i} \sum_d dp(s_i, d) \\ &= \sum_i \frac{1}{1 - a_{ii}^c} p(m_c^t = s_i). \end{aligned} \tag{3}$$

Besides two transition probabilities, the overall expected duration provides an alternative way to analyse the dynamic behaviour. The state distribution (occupied, free) of one point may change with time and the overall expected duration at different times may be different. For convenience, the overall expected duration at a stationary state is applied and given by:

$$E(d) = \mathcal{G}_1 \frac{1}{1 - a_{11}^c} + \mathcal{G}_2 \frac{1}{1 - a_{22}^c}, \tag{4}$$

where \mathcal{G}_1 and \mathcal{G}_2 are the occupancy and free probabilities of the stationary distribution, respectively. The stationary distribution can be obtained by solving:

$$\begin{cases} [\mathcal{G}_1 \ \mathcal{G}_2] \begin{bmatrix} a_{11}^c & 1 - a_{11}^c \\ 1 - a_{22}^c & a_{22}^c \end{bmatrix} = [\mathcal{G}_1 \ \mathcal{G}_2], \\ \mathcal{G}_1 + \mathcal{G}_2 = 1. \end{cases} \tag{5}$$

As a result, the probabilities \mathcal{G}_1 and \mathcal{G}_2 are given by:

$$\mathcal{G}_1 = \frac{1 - a_{22}^c}{2 - a_{11}^c - a_{22}^c}, \tag{6}$$

$$\mathcal{G}_2 = \frac{1 - a_{11}^c}{2 - a_{11}^c - a_{22}^c}. \tag{7}$$

Given two transition probabilities a_{11}^c and a_{22}^c , the overall expected duration can be computed to indicate how dynamic one point is. High dynamic points will have short overall expected duration.

3.2. Parameter Estimation

Since it is not possible to estimate transition probabilities a_{11}^c and a_{22}^c by maximising likelihood directly, the EM algorithm can be applied to estimate the transition probabilities and the initial state probabilities. The parameters are denoted by $\theta_c = \{\rho_i^c, a_{ij}^c\}$, where ρ_i^c represents the initial state probability $p(m_c^0 = s_i)$. Assuming an observation sequence is denoted by $O = (y_0, y_1, \dots, y_{\zeta-1})$ and an underlying state sequence is denoted by

$\mathcal{M}_c = (m_c^0, m_c^1, \dots, m_c^{\zeta-1})$, the likelihood function of θ_c given the observation sequence and the underlying state sequence is:

$$p(O, \mathcal{M}_c | \theta_c) = p(m_c^0)p(z^0 | m_c^0) \prod_{t=1}^{\zeta-1} p(m_c^t | m_c^{t-1})p(y_t | m_c^t). \tag{8}$$

However, an observation sequence y_t does not include all of the space. When m_c^t is not observed, the emission probabilities $p(y_t | m_c^t)$ are set to 1. In order to estimate the parameters, the EM algorithm is applied to recursively maximise a Q function given by:

$$\begin{aligned} Q(\theta_c, \theta_c^{(k)}) &= \sum_{\mathcal{M}_c} p(\mathcal{M}_c | O, \theta_c^{(k)}) \log p(\mathcal{M}_c, O | \theta_c) \tag{9} \\ &= \sum_{\mathcal{M}_c} p(\mathcal{M}_c | O, \theta_c^{(k)}) (\log p(O | \mathcal{M}_c, \theta_c) + \log p(\mathcal{M}_c | \theta_c)) \\ &= \sum_{\mathcal{M}_c} p(\mathcal{M}_c | O, \theta_c^{(k)}) \log p(O | \mathcal{M}_c, \theta_c) \\ &+ \sum_{\mathcal{M}_c} p(\mathcal{M}_c | O, \theta_c^{(k)}) \log p(\mathcal{M}_c | \theta_c). \tag{10} \end{aligned}$$

The sum is over all the possible state sequence \mathcal{M}_c . Since the observation sequence O is conditionally independent of the parameters θ_c given the state sequence \mathcal{M}_c , the probability $p(O | \mathcal{M}_c, \theta_c)$ can be rewritten as $p(O | \mathcal{M}_c)$ and is a constant. As a consequence, the first term in Equation (10) is a constant and the parameters can be estimated by maximising the second term rewritten as:

$$\begin{aligned} &\sum_{\mathcal{M}_c} p(\mathcal{M}_c | O, \theta_c^{(k)}) \log p(\mathcal{M}_c | \theta_c) \\ &= \sum_{i=1}^2 \gamma_i^c(0) \log \rho_i^c + \sum_t \sum_{i=1}^2 \sum_{j=1}^2 \zeta_{ij}^c(t) \log a_{ij}^c \\ &= \sum_{i=1}^2 \gamma_i^c(0) \log \rho_i^c + \sum_t \sum_{i=1}^2 \zeta_{1i}^c(t) \log a_{1i}^c + \sum_t \sum_{i=1}^2 \zeta_{2i}^c(t) \log a_{2i}^c \\ &= f(\rho_1^c) + f(a_{11}^c) + f(a_{22}^c), \tag{11} \end{aligned}$$

where three functions $f(\rho_1^c)$, $f(a_{11}^c)$, and $f(a_{22}^c)$ are defined by:

$$f(\rho_1^c) = \sum_{i=1}^2 \gamma_i^c(0) \log \rho_i^c \tag{12}$$

$$= \gamma_1^c(0) \log \rho_1^c + \gamma_2^c(0) \log(1 - \rho_1^c), \tag{13}$$

$$f(a_{11}^c) = \sum_t \sum_{i=1}^2 \zeta_{1i}^c(t) \log a_{1i}^c \tag{14}$$

$$= \sum_t \zeta_{11}^c(t) \log a_{11}^c + \sum_t \zeta_{12}^c(t) \log(1 - a_{11}^c), \tag{15}$$

$$f(a_{22}^c) = \sum_t \sum_{i=1}^2 \zeta_{2i}^c(t) \log a_{2i}^c \tag{16}$$

$$= \sum_t \zeta_{21}^c(t) \log(1 - a_{22}^c) + \sum_t \zeta_{22}^c(t) \log a_{22}^c. \tag{17}$$

The variable $\gamma_i^c(t)$ represents $p(m_c^t = s_i | O, \theta_c^{(k)})$, which is the probability of being in state s_i at time t given the observation sequence O and the parameters $\theta_c^{(k)}$. The variable $\xi_{ij}^c(t)$ represents $p(m_c^t = s_i, m_c^{t+1} = s_j | O, \theta_c^{(k)})$ which is the probability of being in state s_i at time t and state s_j at time $t + 1$ given the observation sequence O and the parameters $\theta_c^{(k)}$. The three functions contain different parameters and can be maximised individually. Maximising the three functions gives the estimations of the initial state probabilities ρ_i^c and the transition probabilities a_{ii}^c ,

$$\rho_i^{c(k+1)} = \gamma_i^c(0), \tag{18}$$

$$a_{ii}^{c(k+1)} = \frac{\sum_{t=1}^{\zeta-1} \xi_{ii}^c(t)}{\sum_{t=1}^{\zeta-1} \gamma_i^c(t)}. \tag{19}$$

Computing the probabilities $\gamma_i^c(t)$ and $\xi_{ij}^c(t)$ requires temporary variables $\alpha_i^c(t)$ and $\beta_i^c(t)$. The variable $\alpha_i^c(t) = p(y_0, y_1, \dots, y_t, m_c^t = s_i | \theta)$ is the probability of seeing the y_0, y_1, \dots, y_t and being in state s_i at time t . This step, called forward procedure, is computed recursively from time 0 to t as:

$$\alpha_i^c(0) = \rho_i p(y_0 | m_c^0 = s_i), \tag{20}$$

$$\alpha_j^c(t+1) = p(y_{t+1} | m_c^t = s_j) \sum_{i=1}^2 \alpha_i^c(t) a_{ij}^c. \tag{21}$$

The variable $\beta_i^c(t) = p(y_{t+1}, \dots, y_{\zeta-1} | m_c^t = s_i, \theta)$ is the probability of the ending partial sequence $y_{t+1}, \dots, y_{\zeta-1}$ given starting state s_i at time t . This step, called backward procedure, is calculated from time $\zeta - 1$ to t as:

$$\beta_i^c(\zeta - 1) = 1, \tag{22}$$

$$\beta_i^c(t) = \sum_{j=1}^2 \beta_j^c(t+1) a_{ij}^c p(y_{t+1} | m_c^t = s_j). \tag{23}$$

According to Bayes rule, the variables $\gamma_i(t)$ and $\xi_{ij}(t)$ are given as:

$$\gamma_i^c(t) = \frac{\alpha_i^c(t) \beta_i^c(t)}{\sum_{j=1}^2 \alpha_j^c(t) \beta_j^c(t)}, \tag{24}$$

$$\xi_{ij}^c(t) = \frac{\alpha_i^c(t) a_{ij}^c \beta_j^c(t+1) b_{jy_{t+1}}}{\sum_{i=1}^2 \sum_{j=1}^2 \alpha_i^c(t) a_{ij}^c \beta_j^c(t+1) b_{jy_{t+1}}}. \tag{25}$$

During one time instant, the map is assumed to be static. Occupancy grid mapping [30] can be applied to build a temporal occupancy map for every time instant. Since the transition matrix is unknown, the state probabilities $p(m_c^t = s_1)$ and $p(m_c^t = s_2)$ are also unknown. All the state probabilities are temporarily set to 0.5 and the posterior probabilities $p(m_c^t = s_1 | y_t)$ and $p(m_c^t = s_2 | y_t)$ can be obtained. However, the probabilities $p(y_t | m_c^t = s_1)$ and $p(y_t | m_c^t = s_2)$ are required for the HMM. Based on Bayes rule, the posterior probability of the occupied state is given by:

$$p(m_c^t = s_1 | y_t) = \frac{p(y_t | m_c^t = s_1) p(m_c^t = s_1)}{p(y_t)}, \tag{26}$$

where $p(y_t)$ is a constant and $p(m_c^t = s_1)$ is the occupancy probability. Similarly, the posterior probability of the free state is given by:

$$p(m_c^t = s_2 | y_t) = \frac{p(y_t | m_c^t = s_2) p(m_c^t = s_2)}{p(y_t)}, \tag{27}$$

where $p(m_c^t = s_2)$ is the free probability. Since the state probabilities $p(m_c^t)$ are set to 0.5, the probabilities $p(m_c^t = s_1 | y_t)$ and $p(m_c^t = s_2 | y_t)$ are the normalised version of $p(y^t | m_c^t = s_1)$ and $p(y^t | m_c^t = s_2)$. Replaced by $p(m_c^t | y_t)$ to compute $\alpha_i^c(t)$ and $\beta_i^c(t)$ give another two temporary variables:

$$\hat{\alpha}_i^c(t) = \eta_\alpha^c(t)\alpha_i^c(t), \tag{28}$$

$$\hat{\beta}_i^c(t) = \eta_\beta^c(t)\beta_i^c(t), \tag{29}$$

where $\eta_\alpha^c(t)$ and $\eta_\beta^c(t)$ are constants. Using these two new variables directly to calculate $\gamma_i^c(t)$ and $\zeta_{ij}^c(t)$ as in Equations (24) and (25), the same $\gamma_i^c(t)$ and $\zeta_{ij}^c(t)$ can be obtained,

$$\gamma_i^c(t) = \frac{\hat{\alpha}_i^c(t)\hat{\beta}_i^c(t)}{\sum_{j=1}^g \hat{\alpha}_j^c(t)\hat{\beta}_j^c(t)}, \tag{30}$$

$$\zeta_{ij}^c(t) = \frac{\hat{\alpha}_i^c(t)a_{ij}^c\hat{\beta}_j^c(t+1)p(m_c^{t+1} = s_j | y^{t+1})}{\sum_{i=1}^g \sum_{j=1}^g \hat{\alpha}_i^c(t)a_{ij}^c\hat{\beta}_j^c(t+1)p(m_c^{t+1} = s_j | y^{t+1})}. \tag{31}$$

Finally these constants are cancelled. As a result, the probabilities $p(m_c^t | y_t)$ can be used to estimate the parameters conveniently instead of $p(y_t | m_c^t)$. For the case that one point is not observed during one time instant, setting the corresponding emission probabilities $p(y_t | m_c^t)$ to 0.5 also gives the same result.

4. GRF-Based HMM with Known Poses

In the previous section, each point is associated with two HMM parameters: a_{11}^c and a_{22}^c . In this section, GRFs are applied to consider the dependence between the parameters of different points, and the two parameters of one point are assumed to be independent. This means there will be two GRFs, one for each parameter. Given some training points in observed space, the parameters of any test point in continuous space can be predicted.

In the previous section, a grid cell is defined in order to obtain the corresponding observation sequence. In this section, the space is divided into grid cells and the central point of each observed grid cell is chosen as a training point. Meanwhile, the test points are chosen arbitrarily. The coordinate set of training points and test points are denoted by \mathcal{I} and \mathcal{I}_* , respectively. The parameters of training points are denoted by $A = (a_{11}, a_{22})$, where $a_{11} = [\dots, a_{11}^c, \dots]^T (c \in \mathcal{I})$ and $a_{22} = [\dots, a_{22}^c, \dots]^T (c \in \mathcal{I})$. The parameters of test points are denoted by $A_* = (a_{11}^*, a_{22}^*)$, where $a_{11}^* = [\dots, a_{11}^c, \dots]^T (c \in \mathcal{I}_*)$ and $a_{22}^* = [\dots, a_{22}^c, \dots]^T (c \in \mathcal{I}_*)$. In probabilistic form, the parameter estimation of all the selected points including training and test points is:

$$p(A, A_* | O). \tag{32}$$

The problem can be factorised as:

$$p(A, A_* | O) = p(A_* | A)p(A | O), \tag{33}$$

where $p(A | O)$ is an HMM parameter estimation problem for training points and $p(A_* | A)$ is an HMM parameter prediction problem for test points. The parameter estimation for training and test points are done individually.

4.1. HMM Parameter Estimation

Due to the independence between a_{11} and a_{22} , the prior distribution can be factorised as:

$$p(A) = p(a_{11})p(a_{22}), \tag{34}$$

where $p(\mathbf{a}_{11})$ and $p(\mathbf{a}_{22})$ are assumed to have the same distribution. The parameter vector \mathbf{a}_{11} is taken as an example. The log odds form of a_{11}^c is defined as [31]:

$$l_{a_{11}}^c = \log \frac{a_{11}^c}{1 - a_{11}^c}. \tag{35}$$

The vector of all the $l_{a_{11}}^c$ of training points is denoted by $\mathbf{l}_{a_{11}} = [\dots, l_{a_{11}}^c, \dots]^T (c \in \mathcal{I})$ and can be expressed as:

$$\mathbf{l}_{a_{11}} = \log \frac{\mathbf{a}_{11}}{\mathbf{1} - \mathbf{a}_{11}}, \tag{36}$$

where $\mathbf{1}$ is a column of ones and the division is elementwise. The vector $\mathbf{l}_{a_{11}}$ is assumed to be Gaussian distributed with mean vector $\boldsymbol{\mu}_1$ and covariance matrix $\mathbf{K}_{\mathcal{I}\mathcal{I}}$,

$$\mathbf{l}_{a_{11}} \sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{K}_{\mathcal{I}\mathcal{I}}). \tag{37}$$

The covariance function is the Ornstein–Uhlenbeck kernel function as:

$$\mathcal{C}(c, c') = \sigma_f^2 \exp\left(-\frac{|c - c'|}{\ell}\right), \tag{38}$$

where σ_f^2 is the signal variance, the parameter ℓ is the length-scale, and the variables c and c' are the corresponding coordinates of two random variables. The prior distribution $p(\mathbf{a}_{11})$ is:

$$p(\mathbf{a}_{11}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{K}_{\mathcal{I}\mathcal{I}}|}} \exp\left(-\frac{1}{2} U(\mathbf{a}_{11})\right), \tag{39}$$

where n is the number of training points and:

$$U(\mathbf{a}_{11}) = \left(\log \frac{\mathbf{a}_{11}}{\mathbf{1} - \mathbf{a}_{11}} - \boldsymbol{\mu}_1\right)^T \mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1} \left(\log \frac{\mathbf{a}_{11}}{\mathbf{1} - \mathbf{a}_{11}} - \boldsymbol{\mu}_1\right). \tag{40}$$

Similarly, the log odds form of a_{22}^c is defined as:

$$l_{a_{22}}^c = \log \frac{a_{22}^c}{1 - a_{22}^c}. \tag{41}$$

The vector of all the $l_{a_{22}}^c$ of training points is denoted by $\mathbf{l}_{a_{22}} = [\dots, l_{a_{22}}^c, \dots]^T (c \in \mathcal{I})$ and also assumed to be Gaussian distributed. The prior distribution $p(\mathbf{a}_{22})$ is given in the same way by:

$$p(\mathbf{a}_{22}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{K}_{\mathcal{I}\mathcal{I}}|}} \exp\left(-\frac{1}{2} U(\mathbf{a}_{22})\right), \tag{42}$$

where

$$U(\mathbf{a}_{22}) = \left(\log \frac{\mathbf{a}_{22}}{\mathbf{1} - \mathbf{a}_{22}} - \boldsymbol{\mu}_2\right)^T \mathbf{K}_{\mathcal{I}\mathcal{I}}^{-1} \left(\log \frac{\mathbf{a}_{22}}{\mathbf{1} - \mathbf{a}_{22}} - \boldsymbol{\mu}_2\right), \tag{43}$$

and $\boldsymbol{\mu}_2$ is the mean vector of $\mathbf{l}_{a_{22}}$.

During one time instant t , the map that consists of the states of training points is denoted by $\mathcal{M}^t = \{\dots, m_c^t, \dots\} (c \in \mathcal{I})$ and the states of different training points are assumed to be independent. The distribution of the map is:

$$p(\mathcal{M}^t) = \prod_{c \in \mathcal{I}} p(m_c^t). \tag{44}$$

A map sequence consists of different maps \mathcal{M}^t in time and is denoted by $\mathcal{M} = \{\mathcal{M}^0, \dots, \mathcal{M}^t, \dots\}$. As shown in Figure 3, the map sequence depends on space and time. Meanwhile, the map sequence also consists of different state sequences $\mathcal{M}_c (c \in \mathcal{I})$ in space and can also be expressed as $\mathcal{M} = \{\dots, \mathcal{M}_c, \dots\} (c \in \mathcal{I})$. Due to the state independence

between different points, the current state of one point only depends on its previous state. The corresponding distribution is computed by:

$$p(\mathcal{M}) = \prod_{c \in \mathcal{I}} p(\mathcal{M}_c). \tag{45}$$

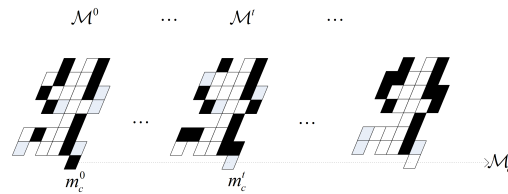


Figure 3. An example of a map sequence.

The likelihood of A given the observation sequence O and the underlying map sequence \mathcal{M} is:

$$p(O, \mathcal{M} | A) = p(O | \mathcal{M}, A)p(\mathcal{M} | A) \tag{46}$$

$$= p(O | \mathcal{M})p(\mathcal{M} | A). \tag{47}$$

Given the map sequence \mathcal{M} , the observation sequence O is conditionally independent of the parameters A . Assuming the measurements are independent of each other, the probability $p(O | \mathcal{M})$ can be obtained by:

$$p(O | \mathcal{M}) = \prod_t p(y_t | \mathcal{M}^t). \tag{48}$$

The measurement sequence y_t only depends on the map configuration \mathcal{M}^t in the same time instant t . The probability $p(y_t | \mathcal{M}^t)$ can be derived from the sensor model,

$$p(y_t | \mathcal{M}^t) = \prod_i p(z_t^i | \mathcal{M}^t). \tag{49}$$

Due to the state dependence between different points, the state sequence \mathcal{M}_c only depends on the corresponding parameter A_c at the same coordinate c and $p(\mathcal{M} | A)$ can be factorised as:

$$p(\mathcal{M} | A) = \prod_{c \in \mathcal{I}} p(\mathcal{M}_c | A_c). \tag{50}$$

The likelihood can be given by:

$$p(O | A) = \sum_{\mathcal{M}} p(O, \mathcal{M} | A). \tag{51}$$

The observation sequence is also conditional on the initial map distribution $p(\mathcal{M}^0)$, which requires the initial state probabilities $\rho_i^c = p(m_c^0 = s_i)$ of training points as in Equation (44). For convenience, the initial probabilities are not written together with A .

Based on Bayes rule, the posterior distribution is:

$$p(A | O) = \frac{p(O | A)p(A)}{p(O)}, \tag{52}$$

where $p(O)$ is a normalising constant.

Similar to the HMM problem in the previous section, the EM algorithm is also applied to estimate the parameters. The Q function with the prior distribution [32] is given as:

$$\begin{aligned}
 Q(\mathbf{A}, \mathbf{A}^{(k)}) &= E_{\mathcal{M}|O, \mathbf{A}^{(k)}} \log p(\mathcal{M}, O | \mathbf{A}) + \log p(\mathbf{A}) \\
 &= E_{\mathcal{M}|O, \mathbf{A}^{(k)}} \log p(O | \mathcal{M}, \mathbf{A}) + E_{\mathcal{M}|O, \mathbf{A}^{(k)}} \log p(\mathcal{M} | \mathbf{A}) \\
 &\quad - 2 \log(\sqrt{(2\pi)^n |K_{\mathcal{I}\mathcal{I}}|}) - \frac{1}{2} U(\mathbf{a}_{11}) - \frac{1}{2} U(\mathbf{a}_{22}),
 \end{aligned} \tag{53}$$

where $\mathbf{A}^{(k)}$ represents the parameters obtained in iteration k . Since the observation sequence O is conditionally independent of the parameters given the map sequence \mathcal{M} , the probabilities $p(O | \mathcal{M}, \mathbf{A})$ can be rewritten as $p(O | \mathcal{M})$ and is a constant. The normalizer, Z , is also a constant. As a result, the parameters can be obtained by maximising the non-constant terms. The second term is rewritten as:

$$\begin{aligned}
 &E_{\mathcal{M}|O, \mathbf{A}^{(k)}} \log p(\mathcal{M} | \mathbf{A}) \\
 &= \sum_{\mathcal{M}} p(\mathcal{M} | O, \mathbf{A}^{(k)}) \sum_{c \in \mathcal{I}} \log p(\mathcal{M}_c | A_c) \\
 &= \sum_{c \in \mathcal{I}} \sum_{\mathcal{M}} p(\mathcal{M} | O, \mathbf{A}^{(k)}) \log p(\mathcal{M}_c | A_c) \\
 &= \sum_{c \in \mathcal{I}} \sum_{\mathcal{M}_c} p(\mathcal{M}_c | O, A_c^{(k)}) \log p(\mathcal{M}_c | A_c).
 \end{aligned} \tag{54}$$

Since the initial state probabilities ρ_i^c are not written together with A_c , the probability $p(\mathcal{M}_c | O, A_c^{(k)})$ is the same as $p(\mathcal{M}_c | O, \theta_c^{(k)})$ in Equation (11), where θ_c includes A_c and the initial state probabilities ρ_i^c . The non-constant terms in this Q function is rewritten as:

$$\begin{aligned}
 &E_{\mathcal{M}|O, \mathbf{A}^{(k)}} \log p(\mathcal{M} | \mathbf{A}) - \frac{1}{2} U(\mathbf{a}_{11}) - \frac{1}{2} U(\mathbf{a}_{22}) \\
 &= \sum_{c \in \mathcal{I}} f(\rho_1^c) + \sum_{c \in \mathcal{I}} f(a_{11}^c) + \sum_{c \in \mathcal{I}} f(a_{22}^c) \\
 &\quad - \frac{1}{2} U(\mathbf{a}_{11}) - \frac{1}{2} U(\mathbf{a}_{22}) \\
 &= f(\boldsymbol{\rho}_1) + f(\mathbf{a}_{11}) + f(\mathbf{a}_{22}),
 \end{aligned} \tag{55}$$

where $\boldsymbol{\rho}_1$ is defined by $\boldsymbol{\rho}_1 = [\dots, \rho_1^c, \dots]^T$ which is the vector of the initial occupancy probabilities ρ_1^c of observed grid cells. The function $f(\boldsymbol{\rho}_1)$, $f(\mathbf{a}_{11})$, and $f(\mathbf{a}_{22})$ are the vector versions of the ones defined in Equation (13), (15), and (17), defined by:

$$\begin{aligned}
 f(\boldsymbol{\rho}_1) &= \sum_{c \in \mathcal{I}} f(\rho_1^c) \\
 &= \gamma_1 \log \boldsymbol{\rho}_1 + \gamma_2 \log(\mathbf{1} - \boldsymbol{\rho}_1),
 \end{aligned} \tag{56}$$

$$\begin{aligned}
 f(\mathbf{a}_{11}) &= \sum_{c \in \mathcal{I}} f(a_{11}^c) - \frac{1}{2} U(\mathbf{a}_{11}) \\
 &= \xi_{11} \log \mathbf{a}_{11} + \xi_{12} \log(\mathbf{1} - \mathbf{a}_{11}) - \frac{1}{2} U(\mathbf{a}_{11}),
 \end{aligned} \tag{57}$$

$$\begin{aligned}
 f(\mathbf{a}_{22}) &= \sum_{c \in \mathcal{I}} f(a_{22}^c) - \frac{1}{2} U(\mathbf{a}_{22}) \\
 &= \xi_{22} \log \mathbf{a}_{22} + \xi_{21} \log(\mathbf{1} - \mathbf{a}_{22}) - \frac{1}{2} U(\mathbf{a}_{22}),
 \end{aligned} \tag{58}$$

where $\gamma_i = [\dots, \gamma_i^c(0), \dots](c \in \mathcal{I})$ and $\xi_{ij} = [\dots, \sum_t \xi_{ij}^c(t), \dots](c \in \mathcal{I})$. The three functions can also be maximised individually. The derivatives of $Q(A, A^{(k)})$ with respect to ρ_1, a_{11} , and a_{22} are, respectively, given by:

$$\frac{d}{d\rho_1} f(\rho_1) = \gamma_1^T \oslash \rho_1 - \gamma_2^T \oslash (\mathbf{1} - \rho_1), \tag{59}$$

$$\begin{aligned} \frac{d}{da_{11}} f(a_{11}) &= \xi_{11}^T \oslash a_{11} - \xi_{12}^T \oslash (\mathbf{1} - a_{11}) \\ &- K_{\mathcal{I}\mathcal{I}}^{-1}(\log \frac{a_{11}}{\mathbf{1} - a_{11}} - \mu_1) \odot \frac{1}{a_{11} \odot (\mathbf{1} - a_{11})}, \end{aligned} \tag{60}$$

$$\begin{aligned} \frac{d}{da_{22}} f(a_{22}) &= \xi_{22}^T \oslash a_{22} - \xi_{21}^T \oslash (\mathbf{1} - a_{22}) \\ &- K_{\mathcal{I}\mathcal{I}}^{-1}(\log \frac{a_{22}}{\mathbf{1} - a_{22}} - \mu_2) \odot \frac{1}{a_{22} \odot (\mathbf{1} - a_{22})}, \end{aligned} \tag{61}$$

where \oslash is the elementwise division and \odot is Hadamard product. Without prior knowledge, the parameter $f(\rho_1)$ can be maximised directly and the estimation of ρ_1 is:

$$\rho_1 = \gamma_1^T. \tag{62}$$

With the prior $p(a_{11})$ and $p(a_{22})$, it is not easy to maximise $f(a_{11})$ and $f(a_{22})$, or equivalently to minimise $-f(a_{11})$ and $-f(a_{22})$. The line search method (LSM) [33] is used to estimate a_{11} and a_{22} in range (0,1). The LSM is a gradient-based method and searches for the optimum of the objective function from the initial values of parameters iteratively. Since this estimation is only one step in the whole optimisation process and only the first iterations are numerically relevant, the LSM can be stopped before convergence in order to achieve computationally efficiency.

4.2. HMM Parameter Prediction

The EM algorithm in the previous section does not give the variances of the parameters a_{11} and a_{22} . After the HMM parameter estimation, all the noise in the observations are assumed to be filtered out and the estimates of a_{11} and a_{22} are assumed to be without noise. Due to the independence between a_{11}^c and a_{22}^c , the prediction problem can be divided as:

$$p(A_* | A) = p(a_{11}^* | a_{11})p(a_{22}^* | a_{22}). \tag{63}$$

Assuming the log odds forms of the parameter vectors a_{11}^* and a_{22}^* for test points are denoted by $l_{a_{11}}^* = [\dots, l_{a_{11}}^c, \dots]^T(c \in \mathcal{I}_*)$ and $l_{a_{22}}^* = [\dots, l_{a_{22}}^c, \dots]^T(c \in \mathcal{I}_*)$, respectively. The distributions $p(a_{11}^* | a_{11})$ and $p(a_{22}^* | a_{22})$ can be derived from $p(l_{a_{11}}^* | l_{a_{11}})$ and $p(l_{a_{22}}^* | l_{a_{22}})$, respectively. The joint distribution of $l_{a_{22}}^*$ and $l_{a_{11}}^*$ is:

$$\begin{bmatrix} l_{a_{11}}^* \\ l_{a_{11}}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_1^* \end{bmatrix}, \begin{bmatrix} K_{\mathcal{I}\mathcal{I}} & K_{\mathcal{I}\mathcal{I}^*}^T \\ K_{\mathcal{I}\mathcal{I}^*} & K_{**} \end{bmatrix}\right), \tag{64}$$

where μ_1^* is the mean vector of $l_{a_{11}}^*$, the matrix $K_{\mathcal{I}\mathcal{I}^*}$ denotes the covariance matrix between $l_{a_{11}}$ and $l_{a_{11}}^*$, and K_{**} is the covariance matrix of $l_{a_{11}}^*$. The predictive equation with noise-free observations [34] is:

$$l_{a_{11}}^* | l_{a_{11}} \sim \mathcal{N}(\bar{l}_{a_{11}}^*, \hat{K}_a^*), \tag{65}$$

where the predictive mean vector $\bar{l}_{a_{11}}^*$ and covariance matrix \hat{K}_a^* are given as:

$$\bar{l}_{a_{11}}^* = \mu_1^* + K_{\mathcal{I}^*} K_{\mathcal{I}\mathcal{I}}^{-1} (l_{a_{11}} - \mu_1), \tag{66}$$

$$\hat{K}_a^* = K_{**} - K_{\mathcal{I}^*} K_{\mathcal{I}\mathcal{I}}^{-1} K_{\mathcal{I}^*}^T. \tag{67}$$

The best predictive parameter vector a_{11}^* of test points is given by the logistic function:

$$a_{11}^* = \frac{\mathbf{1}}{\mathbf{1} + \exp(-\bar{l}_{a_{11}}^*)}, \tag{68}$$

where the division is elementwise. Similarly, the joint distribution of $l_{a_{11}}$ and $l_{a_{11}}^*$ is:

$$\begin{bmatrix} l_{a_{22}} \\ l_{a_{22}}^* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mu_2 \\ \mu_2^* \end{bmatrix}, \begin{bmatrix} K_{\mathcal{I}\mathcal{I}} & K_{\mathcal{I}\mathcal{I}^*}^T \\ K_{\mathcal{I}\mathcal{I}^*} & K_{**} \end{bmatrix} \right), \tag{69}$$

where μ_2^* is the mean vector of $l_{a_{22}}^*$. The coordinates of training points and test points for two prediction problems are the same. As a result, the covariance matrix of the joint distribution and the predictive covariance matrix do not change. The predictive equation of $l_{a_{22}}^*$ is:

$$l_{a_{22}}^* | l_{a_{22}} \sim \mathcal{N}(\bar{l}_{a_{22}}^*, \hat{K}_a^*), \tag{70}$$

where the predictive mean vector $\bar{l}_{a_{22}}^*$ is:

$$\bar{l}_{a_{22}}^* = \mu_2^* + K_{\mathcal{I}^*} K_{\mathcal{I}\mathcal{I}}^{-1} (l_{a_{22}} - \mu_2). \tag{71}$$

The best predictive parameter vector a_{22}^* of test points is:

$$a_{22}^* = \frac{\mathbf{1}}{\mathbf{1} + \exp(-\bar{l}_{a_{22}}^*)}. \tag{72}$$

5. GRF-Based HMM with Pose Uncertainty

The mapping method in the previous section computes the posterior distribution from a prior term and a likelihood term assuming that the precise position of the robot is known. When the uncertainty of robot poses is considered, the problem is how to incorporate that uncertainty into the two terms. The prior distributions depend on the relative positions between different points. However, the chosen points in observed space are the central points of observed grid cells. The points in unobserved space are chosen arbitrarily. As a result, the prior term does not depend on robot poses. Without pose uncertainty, the likelihood term is derived directly from the measurement model $p(z_i | m)$, where m denotes the whole map. When the robot pose \mathcal{P}_k at the time step k is uncertain, the measurement model is $p(z_i | m, \mathcal{P}_k)$. Based on the law of total probability, the uncertainty of the robot pose can be incorporated into the sensor uncertainty by:

$$p(z_i | m) = \int p(z_i | m, \mathcal{P}_k) p(\mathcal{P}_k) d\mathcal{P}_k. \tag{73}$$

In this work, the map is divided into grid cells to obtain observation sequences, however it is not easy to integrate the probability in each grid cell. This problem can be solved by sampling n_s points from the distribution $p(\mathcal{P}_k | \mathcal{Z}_0, \mathcal{Z}_\zeta)$ and averaging over

all the pose samples. Assuming the samples denoted by \mathcal{P}_k^i with associated weights w^i , Equation (73) can be rewritten as:

$$p(z_i | m) = \sum_{i=1}^{n_s} w^i p(z_i | m, \mathcal{P}_k^i). \quad (74)$$

Assuming m denotes a whole grid map and the state of one grid cell is denoted by m_c , the probability $p(z_i | m_c)$ can be obtained by:

$$\begin{aligned} p(z_i | m_c) &= \sum_{m \setminus m_c} p(z_i | m) p(m \setminus m_c | m_c) \\ &= \sum_{m \setminus m_c} \sum_{i=1}^{n_s} w^i p(z_i | m, \mathcal{P}_k^i) p(m \setminus m_c | m_c) \\ &= \sum_{i=1}^{n_s} \sum_{m \setminus m_c} w^i p(z_i | m, \mathcal{P}_k^i) p(m \setminus m_c | m_c) \\ &= \sum_{i=1}^{n_s} w^i p(z_i | m_c, \mathcal{P}_k^i), \end{aligned} \quad (75)$$

where $m \setminus m_c$ means the whole map m without m_c . After the uncertainty of robot poses are incorporated, the proposed method in the previous section can be implemented as usual.

6. Experiments

6.1. Experimental Setup

The experimental platform consists of two parts: The robot part and the PC part. On the robot part, a 3pi robot and one XBee communication module are connected to an mbed expansion board, two IR sensors are connected to the mbed. On the PC part, one XBee module is connected to the PC by an XBee Explorer USB. The 3pi robot is controlled by the mbed microprocessor which sends commands to the robot by a pair of serial ports. The IR sensors 1 and 2 are two Sharp GP2Y0A41SK0F IR sensors, which can measure distances to objects and generate an analog voltage signal. The mbed samples the analog voltages of the IR sensors by two ADC ports. The voltage data can be sent to the PC by XBees 1 and 2, which are XBee S1 802.15.4 low-power modules. The mbed sends data to XBee 1 by another pair of serial ports. XBee 2 receives the data from XBee 1 and sends it to the PC. Since the mbed processor has limited computational power, the PC is in charge of the mapping tasks.

6.2. Experimental Environments

To illustrate the algorithm, the experimental map is shown in Figure 4. There are some objects with different shapes and sizes. The coordinates of the map are shown in Figure 5. The objects with labels 1, 3, 4, 6, 8, and 9 appear and disappear from their positions with different frequencies. The object 1 changes its state at every loop and the subsequent dynamic objects change their states every 2, 5, 10, 20, and 50 loops, respectively. The objects 2, 5, and 7 are static.

A 3pi robot equipped with two Sharp GP2Y0A41SK0F IR sensors is used to test the proposed method. Its diameter is 9.5 cm and the width W between two wheels is 8.2 cm. Two IR sensors are mounted on the robot as shown in Figure 6, and the relative orientations are $\pm 30^\circ$ with respect to the robot's reference frame. The measuring range is 4 to 30 cm. When the distances are more than 20 cm, the output voltage has lower sensitivity and becomes noisier. In the experiments, the maximum distances of IR sensors are set to 20 cm. Since the aperture angle is very small, the two IR sensors do not interface with each other.

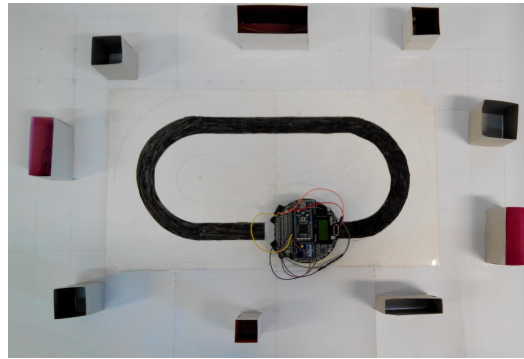


Figure 4. Experimental map.

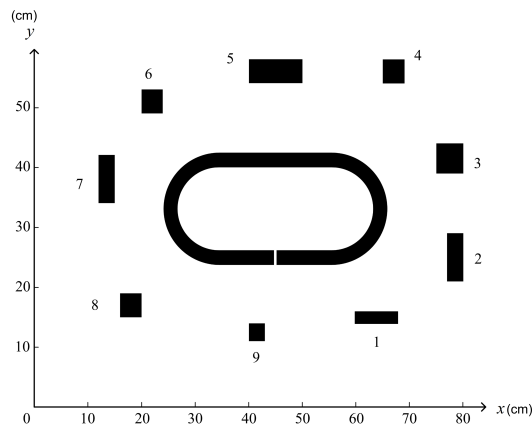


Figure 5. Coordinates of the experimental map.

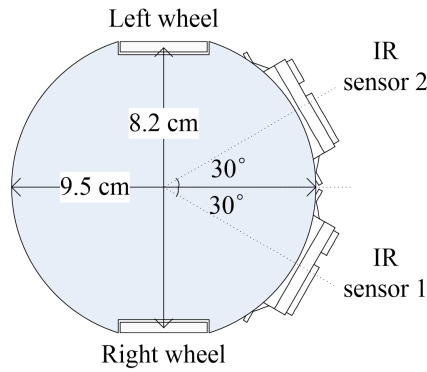


Figure 6. Brief top view of the robot.

6.3. Pose Uncertainty

As the robot explores the environment, its pose uncertainty will increase. A track with a mark is drawn in Figure 4 to decrease the pose uncertainty. Five QTR-RC reflectance sensors in the front of the robot are used to follow the track and detect the mark, and a simple controller is designed to help the robot follow the track. As the robot moves, its pose is predicted by its motion model while the corresponding uncertainty increases. When the robot detects the mark again, the robot closes its trajectory, and all the poses are corrected.

6.3.1. Robot Motion Model

The robot position is represented by the central point between two wheels in world coordinates, and its orientation is relative to the x axis. Its pose vector is denoted by $\mathcal{P} = [x, y, \phi]^T$, which includes the position (x, y) and the orientation ϕ . Assuming the

speeds of the left and right wheels are denoted by v_l and v_r , the speed of the robot is modelled as:

$$\begin{cases} \frac{d}{dt}x = \frac{v_l + v_r}{2} \cos\phi \\ \frac{d}{dt}y = \frac{v_l + v_r}{2} \sin\phi \\ \frac{d}{dt}\phi = \frac{v_r - v_l}{W}. \end{cases} \quad (76)$$

When the robot goes straight shown in Figure 7, its orientation does not change. Assuming the pose at time step $k - 1$ is denoted by $\mathcal{P}_{k-1} = (x_{k-1}, y_{k-1}, \phi_{k-1})$, the next pose \mathcal{P}_k after a time interval Δt can be computed based on the Euler method,

$$\begin{cases} x_k = x_{k-1} + \frac{v_l + v_r}{2} \Delta t \cos\phi_{k-1} \\ y_k = y_{k-1} + \frac{v_l + v_r}{2} \Delta t \sin\phi_{k-1} \\ \phi_k = \phi_{k-1}. \end{cases} \quad (77)$$

When the robot does not move straight shown in Figure 8, direct integration is performed instead of the Euler approximation. Integrating Equation (76) in a time interval Δt gives the new pose [31]:

$$\begin{cases} x_k = x_{k-1} + \frac{W(v_l + v_r)}{2(v_r - v_l)} (\sin(\phi_{k-1} + \frac{v_r - v_l}{W} \Delta t) - \sin\phi_{k-1}) \\ y_k = y_{k-1} + \frac{W(v_l + v_r)}{2(v_r - v_l)} (-\cos(\phi_{k-1} + \frac{v_r - v_l}{W} \Delta t) + \cos\phi_{k-1}) \\ \phi_k = \phi_{k-1} + \frac{v_r - v_l}{W} \Delta t. \end{cases} \quad (78)$$

The system model with additive process noise \mathcal{W} is rewritten as:

$$\mathcal{P}_k = F(\mathcal{P}_{k-1}, v_r, v_l) + \mathcal{W}, \quad (79)$$

where \mathcal{W} is Gaussian distributed with zero mean and covariance matrix \mathcal{R} . The function $F(\mathcal{P}_{k-1}, v_r, v_l)$ represents the expressions on the right sides of Equations (77) and (78).

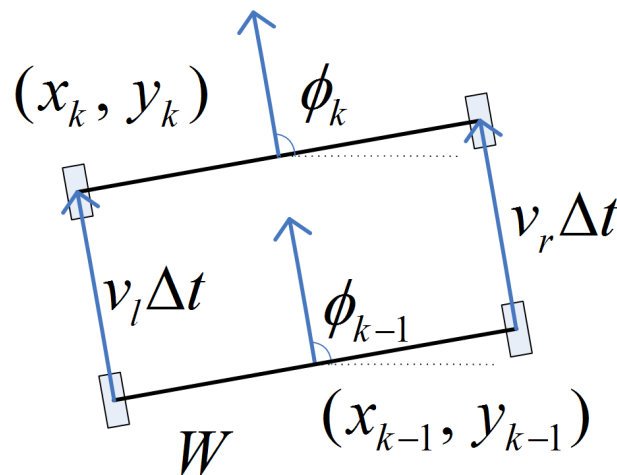


Figure 7. The pose when the robot goes straight.

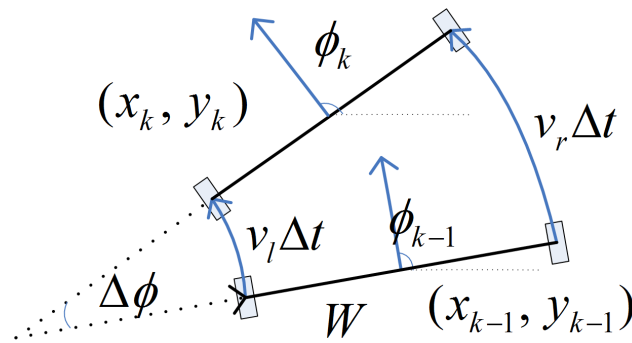


Figure 8. The pose when the robot turns.

6.3.2. Robot Measurement Model

The robot follows the track in a clockwise direction. Once the robot detects the mark, there will be one observation of the position of the robot. The part of the track in front of the mark is straight. When the robot arrives at the mark, the simple controller can adjust the orientation of the robot to be close to $-\pi$. As a result, the position and the orientation become known, although not precisely. However, the observation is not precise. The noisy observation is modelled as:

$$\mathbf{Z}_k = \mathcal{P}_k + [-4.75, 0, 0]^T + \mathcal{V}, \tag{80}$$

where \mathcal{V} is Gaussian noise with zero mean and covariance matrix \mathcal{Q} . The measurement \mathbf{Z}_k is the pose of the robot head where the reflectance sensors are located and \mathcal{P}_k is the pose of the central point of the robot. The pose difference between the head and the central point is $[-4.75, 0, 0]^T$ when the mark is detected.

6.3.3. Pose Smoothing

When the robot starts at the mark and performs a complete loop returning to the mark, only two observations of the pose are available, as shown in Figure 9. The two observations are denoted by \mathbf{Z}_0 and \mathbf{Z}_ζ , respectively. Assuming the position of the mark is at coordinates (x_0, y_0) , both observations \mathbf{Z}_0 and \mathbf{Z}_ζ are equal to $(x_0, y_0, -\pi)$. In order to ensure the smoothness of pose estimation along the trajectory, all the poses along the trajectory should be corrected by the observations. The pose smoothing can be obtained by:

$$p(\mathcal{P}_k | \mathbf{Z}_0, \mathbf{Z}_\zeta) = \frac{p(\mathbf{Z}_\zeta | \mathcal{P}_k)p(\mathcal{P}_k | \mathbf{Z}_0)}{p(\mathbf{Z}_\zeta | \mathbf{Z}_0)}. \tag{81}$$

This formula is divided into two steps: The forward step $p(\mathcal{P}_k | \mathbf{Z}_0)$ and the backward step $p(\mathbf{Z}_\zeta | \mathcal{P}_k)$.

The objective of the forward step is to estimate $p(\mathcal{P}_k | \mathbf{Z}_0)$, which in general can be obtained by iterating the equation:

$$p(\mathcal{P}_k | \mathbf{Z}_0) = \int p(\mathcal{P}_k | \mathcal{P}_{k-1})p(\mathcal{P}_{k-1} | \mathbf{Z}_0)d\mathcal{P}_{k-1}. \tag{82}$$

Since there is no prior knowledge for \mathcal{P}_0 , the posterior distribution $p(\mathcal{P}_0 | \mathbf{Z}_0)$ is assumed to be Gaussian distributed with mean $(x_0, y_0, -\pi)$ and covariance matrix \mathcal{Q} . Based on the previous predicted state distribution $p(\mathcal{P}_{k-1} | \mathbf{Z}_0)$, the current predicted state distribution $p(\mathcal{P}_k | \mathbf{Z}_0)$ can be obtained. Since the robot model is nonlinear and Equation (82) is difficult to integrate, the forward step is done instead by the scaled unscented transformation [35]. The dimension L of the pose vector $p(\mathcal{P}_k)$ is 3 and $2L + 1$ sigma points should be

sampled from the distribution $p(\mathcal{P}_{k-1} | \mathcal{Z}_0)$, which is assumed to be Gaussian distributed with mean vector $\bar{\mathcal{P}}_{k-1}^-$ and covariance matrix \mathbf{P}_{k-1}^- . The sigma points \mathcal{P}_{k-1}^{i-} are given by:

$$\begin{aligned} \mathcal{P}_{k-1}^{0-} &= \bar{\mathcal{P}}_{k-1}^-, \\ \mathcal{P}_{k-1}^{i-} &= \bar{\mathcal{P}}_{k-1}^- + (\sqrt{(L + \lambda)\mathbf{P}_{k-1}^-})_i \text{ for } i = 1, \dots, L, \\ \mathcal{P}_{k-1}^{i-} &= \bar{\mathcal{P}}_{k-1}^- - (\sqrt{(L + \lambda)\mathbf{P}_{k-1}^-})_{i-L} \\ &\text{for } i = (L + 1), \dots, 2L. \end{aligned} \tag{83}$$

The corresponding mean weights w_m^i and covariance weights w_c^i are given by:

$$\begin{aligned} w_m^0 &= \frac{\lambda}{L + \lambda}, \\ w_c^0 &= \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta, \\ w_m^i &= w_c^i = \frac{1}{2(L + \lambda)} \quad i = 1, \dots, 2L, \end{aligned} \tag{84}$$

where α, β , and κ are parameters, the variable $\lambda = \alpha^2(L + \kappa) - L$. The formula $(\sqrt{(L + \lambda)\mathbf{P}_{k-1}^-})_i$ is the i_{th} column of $\sqrt{(L + \lambda)\mathbf{P}_{k-1}^-}$. The parameter α is a positive scaling parameter and its range is $(0,1)$. The parameter κ is a non-negative scaling parameter and its range is $[0, \infty)$. Normally κ is set to 0 [36]. The parameter β is used to incorporate prior knowledge of the distribution of \mathcal{P}_{k-1} . For Gaussian distributions, the optimal choice is 2 [35].

Based on these sigma points, the next pose can be predicted by projecting sigma points through the motion model and new sigma points \mathcal{P}_k^i will be obtained. However, they should be augmented to include the system noise. The first augmented sigma point \mathcal{P}_k^{0-} is the same as \mathcal{P}_k^0 . The other augmented sigma points are given by:

$$\mathcal{P}_k^{i-} = \begin{cases} \mathcal{P}_k^i + (\sqrt{(L + \lambda)\mathbf{R}})_i & \text{for } i = 1, \dots, L, \\ \mathcal{P}_k^i - (\sqrt{(L + \lambda)\mathbf{R}})_{i-L} & \text{for } i = (L + 1), \dots, 2L. \end{cases} \tag{85}$$

The mean vector and covariance matrix of the distribution $p(\mathcal{P}_k | \mathcal{Z}_0)$ can be respectively approximated as:

$$\bar{\mathcal{P}}_k^- \approx \sum_{i=0}^{2L} w_m^i \mathcal{P}_k^{i-}, \tag{86}$$

$$\mathbf{P}_k^- \approx \sum_{i=0}^{2L} w_c^i (\mathcal{P}_k^i - \bar{\mathcal{P}}_k^-)(\mathcal{P}_k^{i-} - \bar{\mathcal{P}}_k^-)^T. \tag{87}$$

The objective of the backward step is to estimate $p(\mathcal{Z}_\zeta | \mathcal{P}_k)$, which in general is given by iterating the equation:

$$p(\mathcal{Z}_\zeta | \mathcal{P}_k) = \int p(\mathcal{Z}_\zeta | \mathcal{P}_{k+1})p(\mathcal{P}_{k+1} | \mathcal{P}_k)d\mathcal{P}_{k+1}. \tag{88}$$

This equation means $p(\mathcal{Z}_\zeta | \mathcal{P}_k)$ can also be obtained recursively based on the robot motion model.

As in the forward step, Equation (88) is difficult to integrate due to the nonlinear characteristics of the motion model, and the same approach based on the unscented transformation is used. The sigma points for the predicted observation are given by:

$$z_\zeta^i = \mathcal{P}_\zeta^i + [-4.75, 0, 0]^T. \tag{89}$$

The mean vector and covariance matrix of the predicted measurement are approximated, respectively, by:

$$\bar{z}_\zeta \approx \bar{\mathcal{P}}_\zeta^- + [-4.75, 0, 0]^T, \tag{90}$$

$$\mathbf{P}_z \approx \mathbf{P}_\zeta^- + \mathbf{Q}. \tag{91}$$

For the pose \mathcal{P}_k , every sigma point \mathcal{P}_k^i corresponds to a sigma point z_ζ^i at the end of the time horizon. The corrected sigma points \mathcal{P}_k^{i+} used to approximate the corrected distribution $p(\mathcal{P}_k | \mathcal{Z}_0, \mathcal{Z}_\zeta)$, which already include the backward step, are computed by:

$$\mathcal{P}_k^{i+} = \mathcal{P}_k^i + \mathcal{K}(\mathcal{Z}_\zeta - z_\zeta^i). \tag{92}$$

The Kalman gain \mathcal{K} and cross variance \mathbf{P}_{kz} are given by:

$$\mathcal{K} = \mathbf{P}_{kz} \mathbf{P}_z^{-1}, \tag{93}$$

$$\mathbf{P}_{kz} = \sum_{i=0}^{2L} w_i^i (\mathcal{P}_k^i - \bar{\mathcal{P}}_k) (z_\zeta^i - \bar{z}_\zeta)^T. \tag{94}$$

The mean vector and covariance matrix of the distribution $p(\mathcal{P}_k | \mathcal{Z}_0, \mathcal{Z}_\zeta)$ are finally estimated as:

$$\begin{aligned} \bar{\mathcal{P}}_k^+ &= \sum_{i=0}^{2L} w_m^i \mathcal{P}_k^{i+} \\ &= \bar{\mathcal{P}}_k^- + \mathcal{K}(\mathcal{Z}_\zeta - \bar{z}_\zeta), \end{aligned} \tag{95}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathcal{K} \mathbf{P}_z \mathcal{K}^T. \tag{96}$$

Taking advantage of corrected sigma points \mathcal{P}_k^{i+} , Equation (74) can be rewritten as:

$$p(z_i | m) = \sum_{i=0}^{2L} w_m^i p(z_i | m, \mathcal{P}_k^{i+}). \tag{97}$$

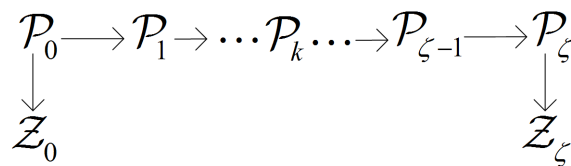


Figure 9. Two observations for the robot.

The position of the mark is set to be at coordinates (45, 25), and the coordinates of the track are shown in Figure 5. The two observations \mathcal{Z}_0 and \mathcal{Z}_ζ are (45, 25, $-\pi$). The covariance matrices \mathcal{R} and \mathbf{Q} are set to:

$$\mathcal{R} = \begin{bmatrix} 10^{-5} & 0 & 0 \\ 0 & 10^{-5} & 0 \\ 0 & 0 & 5 \times 10^{-5} \end{bmatrix},$$

$$\mathbf{Q} = 8 \times 10^{-6} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The other three parameters are set to $\alpha = 0.00001$, $\beta = 2$, and $\kappa = 0$. The predicted position of the robot before reaching the mark is shown in Figure 10, and the ellipses represent the position uncertainty of the robot (level curve of the distribution at three standard deviations). As time goes by, the uncertainty increases. After a complete loop, the

estimated position has drifted to a wrong position, however it is still inside the ellipse. After pose smoothing, the position correction is shown in Figure 11. The estimated positions on the top are out of the outer line of the track with large variances, however the estimated positions near the initial and final positions are more accurate, as expected.

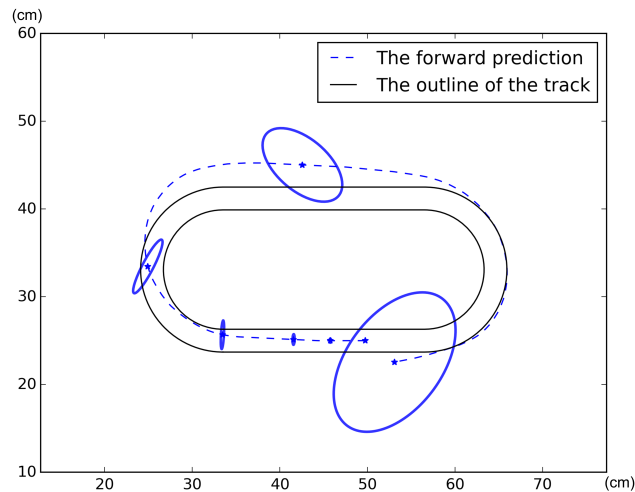


Figure 10. Position prediction.

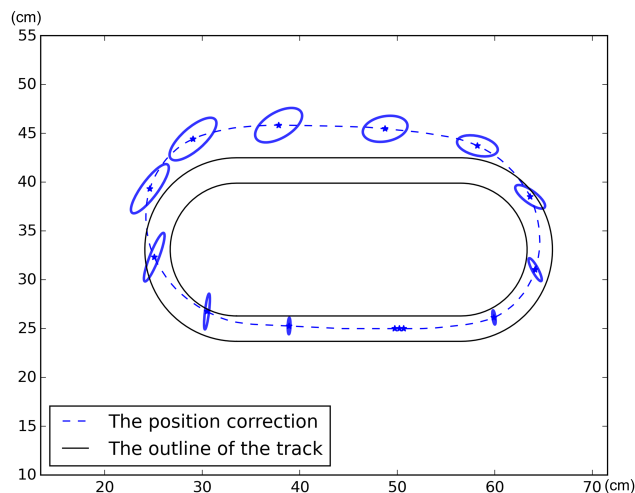


Figure 11. Position correction.

6.4. Results

Regarding the beam model of the two IR sensors, the occupancy and free probabilities $p(z_i | m_c)$ of the occupied grid cells in the measurement ranges are set to 0.998 and 0.002, respectively. The two probabilities of the free grid cells in the measurement ranges are set to 0.008 and 0.992. The two probabilities for grid cells outside the measurement ranges are set to 0.5.

The robot follows the track for 100 loops, and an occupancy grid map is built for every loop. Grid cells with posterior occupancy probabilities larger than 0.5 are assumed to be observed occupied. While posterior occupancy probabilities less than 0.5 are assumed to be observed free. The total times the grid cells are observed free and occupied are shown in Figures 12 and 13, respectively. Most of the space in the middle is observed free many times. The space behind the dynamic objects has about half a chance to be observed, and the space behind the static objects is never observed. For the static objects, their borders are observed partially. Due to the uncertainty of sensors and robot poses, the space around the objects is sometimes observed to be occupied. Similarly to most of the observed space, the number of observed times for dynamic object 6 in Figure 13 is close to 0, possibly due to

the small size of the object combined with the fact that the robot is turning when the range finder crosses the object.

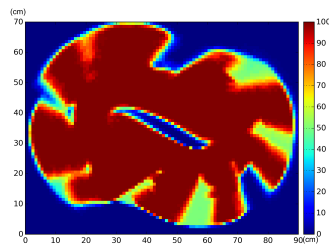


Figure 12. The total number of free observations.

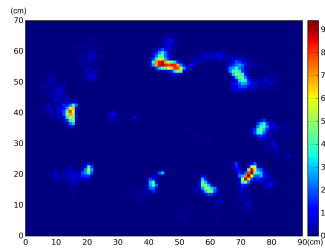


Figure 13. The total number of occupied observations.

The central points of grid cells in observed space are the training data to test the proposed method, and the test points are only chosen from unobserved space. The initial values of the probabilities are set to 0.5, the length-scale ℓ of the covariance function is set to 3, and the signal variance σ_f^2 is set to 25. The mean vectors μ_1 and μ_1^* of the occupied-to-occupied probabilities for the training and test points are set to $\log 9$, which corresponds to a probability of 0.9. The mean vectors μ_2 and μ_2^* are set to $\log 99$, which corresponds to a probability of 0.99. It means the prior knowledge of the environment is slow dynamic. The maximum number of iterations of the optimisation process is set to 800, and the results are shown in Figures 14 and 15, where the unobserved space is covered by asterisks and has no estimates of transition probabilities. For most of the free space, all the observations are free. The parameter a_{11} of most of the observed free space in Figure 14 are close to the initial value 0.5, which should be close to 0. In this area, most of the observations are free. In contrast, the other observed free space with more unknown observations has a better parameter estimate.

In order to discover the reason, several free points without occupied observations are chosen from the observed space and their parameters are individually estimated using the method in Section 3. Table 1 shows the numbers of observations of four free points selected. All of them have no occupied observations and different numbers of free observations. The optimisation processes of their parameters are shown in Figures 16 and 17, where the optimisation processes with more free observations converge faster. The derivatives of $Q(\theta_c, \theta_c^{(k)})$ with respect to a_{11} and a_{22} are shown in Figures 18 and 19, respectively. Even though there is no occupied observation, the derivatives of the Q function with respect to a_{11} are not always zeros and converge to zeros. When the parameters a_{22} converge to 1, the derivatives of Q function with respect to a_{22} converge to nonzero constants. The best estimates of a_{22} for these free points are 1. When a_{22} reaches 0.9, the derivative for corresponding a_{11} is close to 0. For the space without unknown observation, the corresponding parameter a_{11} converge very quickly and a_{11} has lesser chance to be optimised. With different numbers of observations, the estimation processes converge at different speeds.

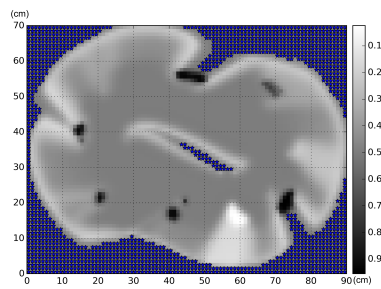


Figure 14. HMM parameter estimation of a_{11} (occupied to occupied).

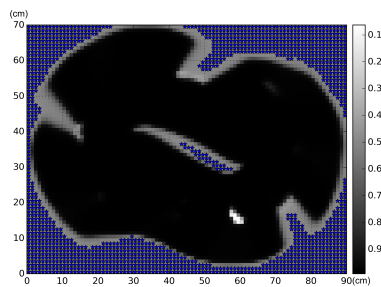


Figure 15. HMM parameter estimation of a_{22} (free to free).

Table 1. Observation numbers of the selected free points.

Free Points	1	2	3	4
Free observation number	19	55	89	100
Occupied observation number	0	0	0	0

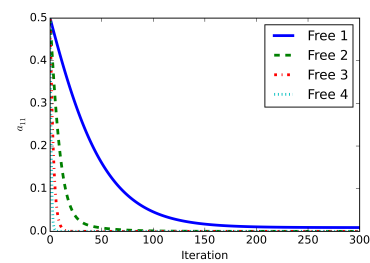


Figure 16. Optimisation process of a_{11} (occupied to occupied).

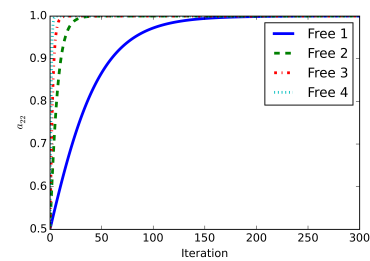


Figure 17. Optimisation process of a_{22} (free to free).

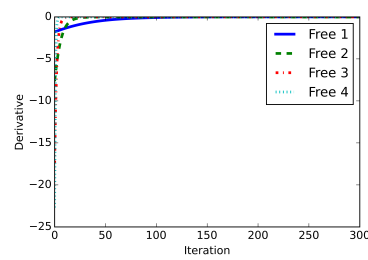


Figure 18. Derivative of $Q(\theta_c, \theta_c^{(k)})$ with respect to a_{11} .

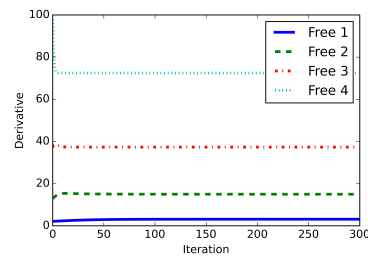


Figure 19. Derivative of $Q(\theta_c, \theta_c^{(k)})$ with respect to a_{22} .

In order to decrease the convergence speed of a_{22} , for the space with more than 95 free observations, 45 of them are replaced by 45 fake observations corresponding to unknown space. the derivative of Q with respect to a_{11} for occupied space and the derivative of Q with respect to a_{22} for free space never converge to zeros. The LSM will always search for the estimates for them even though their estimates are close to 1. In order to give more chances to other parameters, the optimisation process halts searching for them when their estimates reach 0.995. The grid cells with more observations will converge faster. For the border of observed space and unobserved space, there are fewer or no observations, and it will take a long period of time to converge.

Since the points with more observations converge quickly, it will take a long period of time to obtain the best estimates for the points with fewer observations. The maximum number of iterations of the optimisation process is set to 1500. The new results are shown in Figures 20 and 21, where most of the observed free space has low a_{11} and high a_{22} . This means that the state stays free for a long time and changes from occupied to free quickly. The static objects 2, 5, and 7 have the opposite behaviour and parameters. The state of the dynamic object 1 alternates between free and occupied quickly. For the dynamic objects 3 and 4, the colour becomes darker corresponding to slower dynamics. Due to the lack of observations, the dynamic object 6 is estimated as free space. The other two dynamic objects 8 and 9 change their states slowly. The space behind these dynamic objects has fewer observations, the corresponding areas are darker than the space with more free observations. The space behind static objects 2 and 5 is never observed, and therefore there is no estimate. Due to the uncertainty of the robot pose, the space behind the static object 7 has a similar estimate to free space.

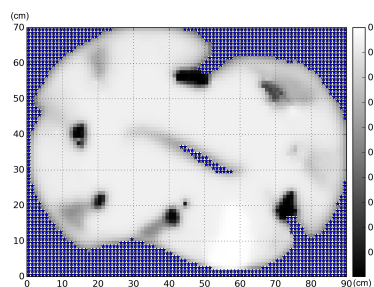


Figure 20. HMM parameter estimation of a_{11} for observed space.

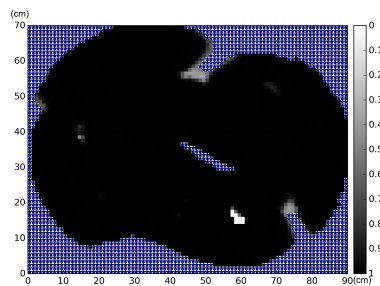


Figure 21. HMM parameter estimation of a_{22} for observed space.

The parameter prediction for unobserved space is shown in Figures 22 and 23. The border of the observed space in Figure 22 is a little fuzzy and the estimate of most of the unobserved space is similar to the prior. In Figure 23, most of the parameters on the borders of observed space are close to 1, and the parameters of the unobserved space near these areas are also predicted to be close to one. Since the parameters of the borders of observed space near the static objects are close to 0.5, the darkness near these areas is lighter. Similarly to Figure 22, the prediction of the other unobserved space is similar to the prior.

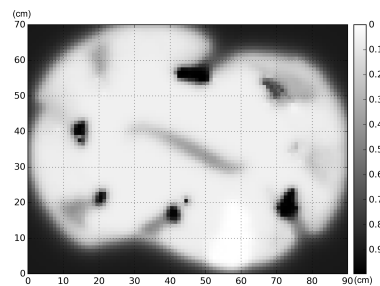


Figure 22. HMM parameter estimation of a_{11} for unobserved space.

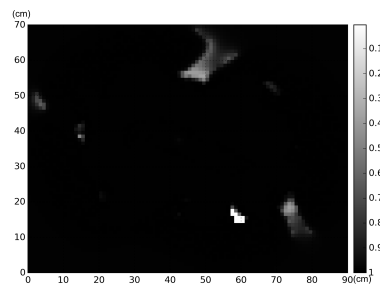


Figure 23. HMM parameter estimation of a_{22} for unobserved space.

Based on parameters a_{11} and a_{22} , the space in dynamic environments can be classified as Table 2. Based on the table, the objects 1, 3, 4, and 6 in the experimental map are high dynamic, and the objects 8 and 9 are low dynamic. The classification results are shown as Figure 24 and Table 3. Due to pose uncertainty, the positions of the objects 8 and 9 are different from the truth. Even though they are obvious in Figure 24, all the True Low Dynamic (TLD) space is wrong. For the same reason, the dynamic objects 3 and 6 are estimated as free space. Due to the prior low dynamic assumption, there is more False Low Dynamic (FLD) space. As the proposed method can smooth the map, there must be more FHD space. The prediction variance increases with the distance to the observed space. As a result, only the predictions near the observed space are more believable. Moreover, the proposed method can predict more free space correctly. The classification accuracy for observed space is 96%.

Table 2. Classification for the dynamic environments. When the space does not belong to any class in the previous four, it is classified as high dynamic.

Classification	Free	Occupied	Low Dynamic	Unknown	High Dynamic
Parameters	$a_{11} < 0.6$ $a_{22} > 0.85$	$a_{11} > 0.85$ $a_{22} < 0.6$	$a_{11} > 0.85$ $a_{22} > 0.85$	$0.53 > a_{11} > 0.47$ $0.53 > a_{22} > 0.47$	Others

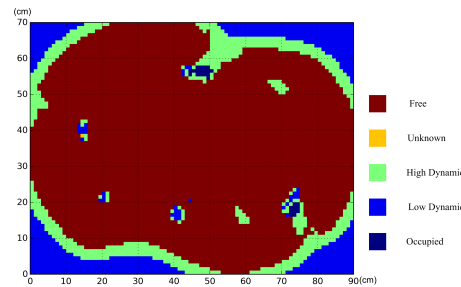


Figure 24. Classification of the results for the dynamic experimental environment.

Table 3. Classification results for the dynamic experimental environment. TF = True free, TO = True occupied, FF = False free, FO = False occupied, TLD = True low dynamic, FLD = False low dynamic, THD = True high dynamic, FHD = False high dynamic, UN = Unknown.

Classification	TF	FF	TO	FO	TLD	FLD	THD	FHD	UN
	4648	146	16	11	0	752	2	725	0

Given the transition probabilities, the overall expected duration is shown in Figure 25 and the version in log scale is shown in Figure 26. The observed free space, the static objects, and the low dynamic objects have long overall expected expectations. The dynamic objects 3 and 6 are mostly invisible and have similar overall expected durations to the free space. The dynamic objects with higher switching frequencies, 1 and 4, have shorter overall expected durations and are clearly visible in log scale. For the unobserved space behind the static objects, the overall expected duration is short. The remaining unobserved space has a long overall expected duration.

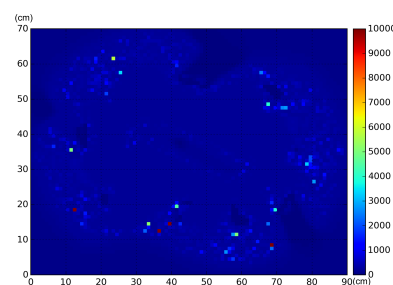


Figure 25. Overall expected duration.

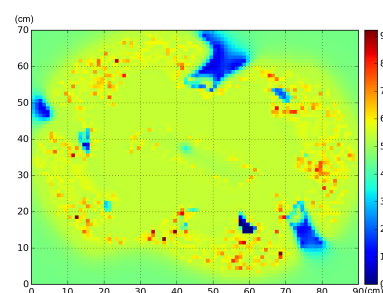


Figure 26. Overall expected duration in log scale.

7. Conclusions

In this paper, a GRF-based mapping method for dynamic environments is proposed, where the dynamic behaviour is modelled by HMMs. The HMM with normalised emission probabilities is introduced and used to conveniently estimate the parameters. In order to deal with the inconsistency in the parameter maps, GRFs are applied to consider the correlations between different points in continue space. The parameter estimation is factorised to reduce computational complexity. The EM algorithm is used to estimate the parameters for the observed space, where the Q function is optimised by the line search method. The predictive equation of the Gaussian process is used to deal with the parameters for the unobserved space. The pose uncertainty is incorporated into the measurement model and a 3pi robot with two IR sensors is used to evaluate the proposed method. Experiment results show that parameter estimation depends on robot poses. Even though the proposed method identifies the low dynamic objects, the number of TLD is 0. The classification accuracy for observed space is 96%. Compared with the state-of-the-art approaches, the proposed method takes the parameter dependence into consideration and builds smooth maps for observed space. Moreover, the dynamic behaviour near the observed space can be predicted, which is significant for path planning. The main disadvantage is that it takes a long period of time to search for the parameters of points with fewer observations. In future work, we plan to reduce the computational complexity.

Author Contributions: Conceptualisation, H.L., M.B. and L.R.; methodology, H.L., M.B. and L.R.; software, H.L., M.B. and L.R.; validation, H.L., M.B. and L.R.; formal analysis, H.L., M.B. and L.R.; investigation, H.L., M.B. and L.R.; resources, H.L., M.B. and L.R.; data curation, H.L., M.B. and L.R.; writing—original draft preparation, H.L., M.B. and L.R.; writing—review and editing, M.B., L.R. and S.W.; visualization, H.L.; supervision, M.B., L.R. and S.W.; project administration, M.B. and L.R.; funding acquisition, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by EACEA under the Erasmus Mundus Action 2, Strand 1 project LEADER and Scientific and Technological Project in Henan Province (grant number 212102210080 and 222102210019).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moravec, H.; Elfes, A. High resolution maps from wide angle sonar. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; pp. 116–121.
2. Plebe, A.; Kooij, J.F.; Papini, G.P.R.; Da Lio, M. Occupancy grid mapping with cognitive plausibility for autonomous driving applications. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2934–2941.
3. Mugnai, F.; Ridolfi, A.; Bianchi, M.; Franchi, M.; Tucci, G. Developing affordable bathymetric analysis techniques using non-conventional payload for cultural heritage inspections. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 807–811. [[CrossRef](#)]
4. Wang, C.-C.; Thorpe, C. Simultaneous localization and mapping with detection and tracking of moving objects. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; pp. 2918–2924.
5. Steyer, S.; Lenk, C.; Kellner, D.; Tanzmeister, G.; Wollherr, D. Grid-based object tracking with nonlinear dynamic state and shape estimation. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2874–2893. [[CrossRef](#)]
6. Schreiber, M.; Belagiannis, V.; Gläser, C.; Dietmayer, K. Dynamic occupancy grid mapping with recurrent neural networks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 6717–6724.
7. Bengio, Y.; Frasconi, P. An input output hmm architecture. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 27–30 November 1995; pp. 427–434.
8. Meyer-Delius, D.; Beinhofer, M.; Burgard, W. Occupancy grid models for robot mapping in changing environments. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22 July 2012; pp. 2024–2030.
9. Luber, M.; Arras, K.O.; Plagemann, C.; Burgard, W. Classifying dynamic objects. *Auton. Robot.* **2009**, *26*, 141–151. [[CrossRef](#)]
10. Tipaldi, G.D.; Meyer-Delius, D.; Burgard, W. Lifelong localization in changing environments. *Int. J. Robot. Res.* **2013**, *32*, 1662–1678. [[CrossRef](#)]

11. Wang, Z.; Ambrus, R.; Jensfelt, P.; Folkesson, J. Modeling motion patterns of dynamic objects by iohmm. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 1832–1838.
12. Dadhich, A.; Koganti, N.; Shibata, T. Modeling occupancy grids using edhmm for dynamic environments. In Proceedings of the 2015 Conference on Advances in Robotics, New York, NY, USA, 2 July 2015; pp. 1–6.
13. Rapp, M.; Dietmayer, K.; Hahn, M.; Duraisamy, B.; Dickmann, J. Hidden markov model-based occupancy grid maps of dynamic environments. In Proceedings of the 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 1780–1788.
14. Rabiner, L.R. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *7*, 257–286. [[CrossRef](#)]
15. Tingdahl, D.; Gool, L.V. A public system for image based 3d model generation. In Proceedings of the International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications, Rocquencourt, France, 10–11 October 2011; pp. 262–273.
16. Li, H.; Barão, M.; Rato, L. Mapping dynamic environments using markov random field models. In Proceedings of the 24th International Conference on Automation and Computing (ICAC), Newcastle Upon Tyne, UK, 6–7 September, 2018; pp. 1–5.
17. O’Callaghan, S.T.; Ramos, F.T. Gaussian process occupancy maps. *Int. J. Robot. Res.* **2012**, *31*, 42–62. [[CrossRef](#)]
18. Kim, S.; Kim, J. Building occupancy maps with a mixture of gaussian processes. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MA, USA, 14–18 May 2012; pp. 4756–4761.
19. Kim, S.; Kim, J. Continuous occupancy maps using overlapping local gaussian processes. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4709–4714.
20. Kim, S.; Kim, J. Recursive bayesian updates for occupancy mapping and surface reconstruction. In Proceedings of the Australasian Conference on Robotics and Automation, Melbourne, Australia, 2–4 December 2014; pp. 1–8.
21. Vido, C.E.; Ramos, F. From grids to continuous occupancy maps through area kernels. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1043–1048.
22. Wang, J.; Englot, B. Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1003–1010.
23. Lee, B.; Zhang, C.; Huang, Z.; Lee, D.D. Online continuous mapping using gaussian process implicit surfaces. In Proceedings of the International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 6884–6890.
24. Ossevorth, F.; Schegner, P. Approximating stochastic loads using the em-algorithm. *IFAC J. Syst. Control* **2021**, *18*, 100175. [[CrossRef](#)]
25. Scaradozzi, D.; Zingaretti, S.; Ferrari, A. Simultaneous localization and mapping (slam) robotics techniques: A possible application in surgery. *Shanghai Chest* **2018**, *2*, 1–11. [[CrossRef](#)]
26. Bibby, C.; Reid, I. Simultaneous localisation and mapping in dynamic environments (slamide) with reversible data association. In Proceedings of the Robotics: Science and Systems, Atlanta, GA, USA, 27–30 June 2007; pp. 105–112.
27. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
28. Rocher, G.; Lavirotte, S.; Tigli, J.-Y.; Cotte, G.; Dechavanne, F. An iohmm-based framework to investigate drift in effectiveness of iot-based systems. *Sensors* **2021**, *21*, 527. [[CrossRef](#)] [[PubMed](#)]
29. Rato, L. Controlo Computado Baseado em Modelos Múltiplos. Ph.D. Thesis, Technical University of Lisbon, Lisbon, Portugal, 2002.
30. Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer* **1989**, *22*, 46–57. [[CrossRef](#)]
31. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: London, UK, 2005; pp. 74–77.
32. Gupta, M.R.; Chen, Y. Theory and use of the em algorithm. *Found. Trends Signal Process.* **2011**, *4*, 223–296. [[CrossRef](#)]
33. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin, Germany, 2006.
34. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; MIT Press: London, UK, 2006; pp. 13–19.
35. Julier, S.J. The scaled unscented transformation. In Proceedings of the American Control Conference, Anchorage, MI, USA, 8–10 May 2002; pp. 4555–4559.
36. Van Der Merwe, R. Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. Ph.D. Thesis, Oregon Health & Science University, Portland, ON, USA, 2004.