SPATIO-TEMPORAL FIELD PREDICTION UNDER LOCALIZATION UNCERTAINTY FOR MOBILE SENSOR NETWORKS

By

Mahdi Jadaliha

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering-Doctor of Philosophy

2013

ABSTRACT

SPATIO-TEMPORAL FIELD PREDICTION UNDER LOCALIZATION UNCERTAINTY FOR MOBILE SENSOR NETWORKS

By

Mahdi Jadaliha

Recently, there has been a growing interest in wireless sensor networks due to the advanced embedded system and network technologies. Their applications include, but are not limited to, environment monitoring, building comfort control, traffic control, manufacturing and plant automation, and surveillance systems. The conventional inverse problem approach based on physical transport models is computationally prohibitive for resource-constrained, multi-agent systems. In contrast, Gaussian process and Gaussian Markov models have been widely used to draw statistical inference from geostatistical and environmental data. However, the statistical models need to be carefully tailored such that they can be practical and usable for mobile sensor networks with limited resources. Additionally, one needs to be careful in coping with localization uncertainty resulted from low-cost mobile sensor networks. Thus, a fundamental problem in various applications is to correctly fuse the spatially collected data and estimate the process of interest under localization uncertainty. Motivated by the aforementioned issues, in this dissertation, we consider the problem of using mobile sensor networks to estimate and predict environmental fields modeled by spatio-temporal Gaussian processes and Gaussian Markov random fields in the presence of localization uncertainty.

In the first part of this dissertation, we formulate Gaussian process regression with observations under the localization uncertainty. In our formulation, effects of measurement noise, localization uncertainty, and prior distributions are all correctly incorporated in the posterior predictive statistics. The analytically intractable posterior predictive statistics are proposed to be approximated by two techniques, viz., Monte Carlo sampling and Laplaces method. In addition, the localization problem is studied in this part, when the position of the robot is estimated by a maximum likelihood estimation (MLE) using vision data. We transform the high dimensional vision data to a set of uncorrelated feature candidates. A multivariate GP regression with unknown hyperparameters is formulated to connect the set of selected features to their corresponding sampling positions. In order to decrease computational load and increase the accuracy of localization, a feature reduction approach is developed. Therefore, a subset of the features is selected to minimize the localization error using cross-validation methods.

In the second part of the dissertation, we consider the problem of predicting a spatial (spatio-temporal) Gaussian Markov random field (GMRF) using sequential noisy observations collected by robotic sensors. The random field of interest is modeled by a GMRF instead of Gaussian process. In this way, we propose iteratively updated predictive inferences. We derive the exact Bayesian solution to the problem of computing the predictive inference of the random field, taking into account uncertain hyperparameters, measurement noise, and uncertain localization in a fully Bayesian point of view. We show that the exact solution is not scalable as the number of observations increases. To cope with this exponentially increasing complexity, we propose scalable approximations with a controllable tradeoff between approximation error and complexity to the exact solution. Finally, we derive an approximate Bayesian solution to the problem of the simultaneously localization and computing the predictive inferences, taking into account observations, uncertain hyperparameters, measurement noise, kinematics of robotic sensors, and uncertain localization.

ACKNOWLEDGMENTS

I would like to thank everybody who helped to accomplish this dissertation. First of all, I would like to express my sincere appreciation and gratitude to my adviser, Dr. Jongeun Choi, who continuously encouraged, and supported me. This thesis is not possible without his invaluable advice and effort to improve the work. In addition, I would like to thank all my committee members, Dr. Steven Shaw, Dr. Guoming Zhu, and Dr. Tapabrata Maiti, for their contribution and suggestions.

Last but not least, I cannot thank enough my sister, my parents and my wife for always being there for me. I would never have been able to accomplish what I have without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Preliminaries

1.1 Mathematical Notation

Standard notation will be used throughout the dissertation. Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{Z}$, $\mathbb{Z}_{\geq 0}$, $\mathbb{Z}_{>0}$ denote, respectively, the sets of real, non-negative real, positive real, integer, non-negative integer, and positive integer numbers. I_n denotes the identity matrix of size n (I for an appropriate dimension).

For column vectors $v_a \in \mathbb{R}^a$, $v_b \in \mathbb{R}^b$, and $v_c \in \mathbb{R}^c$, $\mathrm{col}(v_a, v_b, v_c) := \begin{bmatrix} v_a^T & v_b^T & v_c^T \end{bmatrix}^T \in \mathbb{R}^{a+b+c}$ stacks all vectors to create one column vector, and $\|v_a\|$ denotes the Euclidean norm (or the vector 2-norm) of v_a. Let $\mathrm{row}_i(A) \in \mathbb{R}^m$ and $\mathrm{col}_j(A) \in \mathbb{R}^n$ denote the i-th row and the j-th column of a matrix $A \in \mathbb{R}^{n \times m}$, respectively.

$|A|$ and $|(|A)$ denote the determinant of a matrix $A \in \mathbb{R}^{n \times n}$, and $\mathrm{tr}(A)$ denotes trace of a square matrix A. Let $A^T \in \mathbb{R}^{m \times n}$ be the transpose of a matrix $A \in \mathbb{R}^{n \times m}$. The positive definiteness and the positive semi-definiteness of a square matrix A are denoted by $A \succ 0$ and $A \succeq 0$, respectively. The symbol $\otimes$ denotes the Kronecker product.

Let $\mathbb{E}$, $\mathbb{V}\mathrm{ar}$ and $\mathbb{C}\mathrm{ov}$ denote the expectation operator, the variance operator and the covariance operator, respectively.

A random vector $z \in \mathbb{R}^q$, which is distributed by a multivariate Gaussian distribution of a mean $\mu \in \mathbb{R}^q$ and a variance $\Sigma \in \mathbb{R}^{q \times q}$, is denoted by $z \sim \mathcal{N}(\mu, \Sigma)$.

We define the first and the second derivative operators on $h : \mathbb{R}^m \to \mathbb{R}$ with respect to a vector $x \in \mathbb{R}^m$ as follow.

$$\nabla h(x) = \frac{\partial h(x)}{\partial x} = \left(\frac{\partial h(x)}{\partial x_1}, \cdots, \frac{\partial h(x)}{\partial x_m} \right)^T,$$

$$\nabla^2 h(x) = \frac{\partial^2 h(x)}{\partial x \partial x^T} = \begin{pmatrix} \frac{\partial^2 h(x)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 h(x)}{\partial x_1 \partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 h(x)}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2 h(x)}{\partial x_m \partial x_m} \end{pmatrix}.$$

Let $\|x\|$ denote the standard Euclidean norm (2-norm) of a vector x. If there exists c and $k \in \mathbb{R}_{>0}$, such that the approximation error satisfies $\|\hat{x} - x\| \leq k|\epsilon|, \forall \epsilon < c$, we say that the error between $\hat{x}$ and x is of order $\mathcal{O}(\epsilon)$ and also write $\hat{x} - x = \mathcal{O}(\epsilon)$.

The relative complement of a set $\mathcal{A}$ in a set $\mathcal{B}$ is denoted by $\mathcal{B} \setminus \mathcal{A} := \mathcal{B} \cap \mathcal{A}^c$, where $\mathcal{A}^c$ is the complement of $\mathcal{A}$. For a set $\mathcal{A} \in \mathcal{I}$, we define $z_{\mathcal{A}} = \{z_i \,|\, i \in \mathcal{A}\}$. Let $-\mathcal{A}$ denote the set $\mathcal{I} \setminus \mathcal{A}$. For given $\mathcal{A} = \{c, d\}$ and $\mathcal{B} = \{1, 2\}$, the multiplication between two sets is defined as $\mathcal{B} \times \mathcal{A} = \{(1, c), (1, d), (2, c), (2, d)\}$. Other notation will be explained in due course.

1.2 Physical Process Model

In this section, we review important notions on the Gaussian process which will be used to model the physical phenomenon. In particular, we introduce a class of spatio-temporal Gaussian process model with anisotropic covariance functions. The properties of Gaussian Markov Random fields (GMRF) are also briefly reviewed.

1.2.1 Gaussian process

A Gaussian process can be thought of a generalization of a Gaussian distribution over a finite vector space to function space of infinite dimension. It is formally defined as follows [83,84]:

Definition 1.2.1. *A Gaussian process (GP) is a collection of random variables, any finite number of which have a consistent*[1] *joint Gaussian distribution.*

A Gaussian process

$$z(x) \sim \mathcal{GP}\left(\mu(x), \mathcal{K}(x, x'; \theta)\right) \tag{1.1}$$

is completely specified by its mean function $\mu(x)$ and covariance function $\mathcal{K}(x, x'; \theta)$. Although not needed to be done, we take the mean function to be zero for notational simplicity[2], *i.e.*, $\mu(x) = 0$. If the covariance function $\mathcal{K}(x, x'; \theta)$ is invariant to translations in the input space, *i.e.*, $\mathcal{K}(x, x'; \theta) = \mathcal{K}(x - x'; \theta)$, we call it stationary. Furthermore, if the covariance function is a function of only the distance between the inputs, *i.e.*, $\mathcal{K}(x, x'; \theta) = \mathcal{K}(\left\|x - x'\right\|; \theta)$, then it is called isotropic.

In practice, a parametric family of functions is used instead of fixing the covariance function [3]. One common choice of a stationary covariance function is

$$\mathcal{K}(x, x'; \theta) = {\sigma_f}^2 \exp\left\{ -\sum_{\ell=1}^{b} \frac{\left(x_\ell - x'_\ell\right)^2}{2\sigma_\ell^2} \right\}, \tag{1.2}$$

where x_ℓ is the ℓ-th element of $x \in \mathbb{R}^b$. From (1.2), it can be easily seen that the correlation between two inputs decreases as the distance between them increases. This decreasing rate

[1] It is also known as the marginalization property [83]. It means simply that the random variables obey the usual rules of marginalization, etc.

[2] This is not a drastic limitation since the mean of the posterior process is not confined to zero [84].

depends on the choice of the kernel bandwidths $\{\sigma_\ell\}$. A very large kernel bandwidth means that the predictions would have little bearing on the corresponding input which is then said to be insignificant. ${\sigma_f}^2$ gives the overall vertical scale relative to the mean of the Gaussian process in the output space. These parameters play the role of hyperparameters since they correspond to the hyperparameters in neural networks and in the standard parametric model. Therefore, we define $\theta = ({\sigma_f}^2, \sigma_1, \cdots, \sigma_b)^T \in \mathbb{R}^{b+1}$ as the hyperparameter vector.

1.2.2 Spatio-temporal Gaussian process

In the first part of this dissertation, spatio-temporal Gaussian processes are of particular interest. Consider a spatio-temporal Gaussian process

$$z(s,t) \sim \mathcal{GP}(\mu(s,t), \mathcal{K}(s,t,s',t';\theta)),$$

which is a special case of the Gaussian process defined in (1.1), where $x = (s^T, t)^T \in \mathbb{R}^\iota \times \mathbb{R}_{\geq 0}$. We consider the following generalized anisotropic covariance function $\mathcal{K}(x, x'; \theta)$ with a hyperparameter vector $\theta := ({\sigma_f}^2, \sigma_1, \cdots, \sigma_\iota, \sigma_t)^T \in \mathbb{R}^{\iota+2}$:

$$\mathcal{K}(x, x'; \theta) = {\sigma_f}^2 \exp\left(-\sum_{\ell=1}^{\iota} \frac{(s_\ell - s'_\ell)^2}{2\sigma_\ell^2}\right) \exp\left(-\frac{(t-t')^2}{2\sigma_t^2}\right), \tag{1.3}$$

where $s, s' \in \mathcal{S} \subset \mathbb{R}^\iota$, $t, t' \in \mathbb{R}_{\geq 0}$. $\{\sigma_1, \cdots, \sigma_\iota\}$ and σ_t are kernel bandwidths for space and time, respectively. (1.3) shows that points close in the measurement space and time indices are strongly correlated and produce similar values. In reality, the larger temporal distance two measurements are taken with, the less correlated they become, which strongly supports our generalized covariance function in (1.3). This may also justify the truncation (or windowing) of the observed time series data to limit the size of the covariance matrix for reducing the computational cost. A spatially isotropic version of the covariance function in (1.3) has been used in [62].

1.2.3 Gaussian Markov random field

In this section, we introduce Gaussian Markov random fields (GMRFs). An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a tuple consisting of a set of vertices $\mathcal{V} := \{1, \cdots, n\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The neighbors of $i \in \mathcal{V}$ in $\mathcal{G}$ are denoted by $N_i := \{j \in \mathcal{V} \mid \{i, j\} \in \mathcal{E}\}$. The GMRF is formally defined as follows [88].

Definition 1.2.2. *(GMRF, [88, Definition* 2.1*]) A random vector* $\boldsymbol{z} = (z^{[1]}, \cdots, z^{[n]})^T$ $\in \mathbb{R}^n$ *is called a GMRF with respect to a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *with mean* μ *and precision matrix* $\boldsymbol{Q} \succ 0$*, if and only if its density has the form*

$$\mathbb{P}(\boldsymbol{z}) = \frac{|\boldsymbol{Q}|^{1/2}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(\boldsymbol{z} - \mu)^T \boldsymbol{Q}(\boldsymbol{z} - \mu)\right),$$

and $(\boldsymbol{Q})_{ij} \neq 0 \Leftrightarrow \{i, j\} \in \mathcal{E}$ *for all* $i \neq j$*, where the precision matrix (or information matrix)* $\boldsymbol{Q} = \boldsymbol{K}^{-1}$ *is the inverse of the covariance matrix* $\boldsymbol{K}$*.*

1.3 Mobile Sensor Network

In this section, we consider the sensor network formed by multiple mobile sensing agents and present the measurement model used throughout the thesis.

Let N be the number of sensing agents distributed over the surveillance region $\mathcal{S} \subset \mathbb{R}^\iota$. The identity of each agent is indexed by $\mathcal{I} := \{1, 2, \cdots, N\}$. Assume that all agents are equipped with identical sensors and take noisy observations at time $t \in \mathbb{Z}_{>0}$. At time t, the sensing agent i takes a noise-corrupted measurement $y^{[i]}(t)$ at its current location $s^{[i]}(t) \in \mathcal{S}$, *i.e.*,

$$y^{[i]}(t) = z(s^{[i]}(t), t) + \epsilon^{[i]}(t), \quad \epsilon^{[i]}(t) \overset{i.i.d.}{\sim} \mathcal{N}(0, {\sigma_\epsilon}^2),$$

where the sensor noise $\epsilon^{[i]}$ is considered to be an independent and identically distributed Gaussian random variable. ${\sigma_\epsilon}^2 > 0$ is the noise level and we define the signal-to-noise ratio as

$$\gamma = \frac{{\sigma_f}^2}{{\sigma_\epsilon}^2}.$$

Notice that when a static field is considered, we have $z(s, t) = z(s)$.

For notational simplicity, we denote the collection of positions of all N agents at time t as $\mathbf{s}_t$, *i.e.*,

$$\mathbf{s}_t := \left(s^{[1]}(t)^T, \cdots, s^{[N]}(t)^T\right)^T \in \mathcal{S}^N.$$

The collective measurements from all N mobile sensors at time t is denoted by

$$\mathbf{y}_t := \left(y^{[1]}(t), \cdots, y^{[N]}(t)\right)^T \in \mathbb{R}^N.$$

The cumulative measurements from time $t \in \mathbb{Z}_{>0}$ to time $t' \in \mathbb{Z}_{>0}$ is denoted by

$$\mathbf{y}_{t:t'} := \left(\mathbf{y}_t^T, \cdots, \mathbf{y}_{t'}^T\right)^T \in \mathbb{R}^{(t'-t+1)N}.$$

1.4 Gaussian Processes for Regression

Suppose we have a data set $\mathcal{D}' = \left\{(x^{[i]}, y^{[i]}) \,|\, i = 1, \cdots, n\right\}$ collected by mobile sensing agents where $x^{[i]}$ denotes an input vector of dimension b and $y^{[i]}$ denotes a scalar value of the noise corrupted output. The objective of probabilistic regression is to compute the predictive distribution of the function values $z^\star := z(x^\star)$ at some test input $x^\star$.

For notational simplicity, we define the collective input vector $\mathbf{x}$ of dimension $n \times b$ as the aggregation of n input vectors (*i.e.*, $\mathbf{x} := (x^{[1]}, \cdots, x^{[n]})^T$), and the outputs are collected in a vector $\mathbf{y} := (y^{[1]}, \cdots, y^{[n]})^T$. The corresponding vector of noise-free outputs is defined as $\mathbf{z} := (z(x^{[1]}), \cdots, z(x^{[n]}))^T$.

The advantage of the Gaussian process formulation is that the combination of the prior and noise models can be carried out exactly via matrix operations [111]. The idea of Gaussian process regression is to place a GP prior directly on the space of functions without parameterizing the function $z(\cdot)$, *i.e.*,

$$\mathbb{P}(\mathbf{z}|\theta) = \mathcal{N}(\mu, \mathbf{K}),$$

where $\mu \in \mathbb{R}^n$ is the mean vector obtained by $\mu_i = \mu(x^{[i]})$, and $\mathbf{K} := \mathbb{C}\text{ov}(\mathbf{z}, \mathbf{z}|\theta) \in \mathbb{R}^{n \times n}$ is the covariance matrix obtained by $(\mathbf{K})_{ij} = \mathcal{K}(x^{[i]}, x^{[j]}; \theta)$. Notice that the GP model and all expressions are always conditional on the corresponding inputs. In the following, we will neglect the explicit conditioning on the input matrix $\mathbf{x}$.

The inference in the Gaussian process model is as follows. First, we assume a joint GP prior $\mathbb{P}(\mathbf{z}, z^\star|\theta)$ over functions, *i.e.*,

$$\mathbb{P}(\mathbf{z}, z^\star|\theta) = \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu(x^\star) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & k \\ k^T & \mathcal{K}(x^\star, x^\star; \theta) \end{bmatrix}\right), \tag{1.4}$$

where $k := \mathbb{C}\mathrm{ov}(\mathbf{z}, z^\star|\theta) \in \mathbb{R}^n$ is the covariance between $\mathbf{z}$ and $z^\star$ obtained by $k_i = \mathcal{K}(x^{[i]}, x^\star; \theta)$. Then, the joint posterior is obtained using Bayes rule, *i.e.*,

$$\mathbb{P}(\mathbf{z}, z^\star|\theta, \mathbf{y}) = \frac{\mathbb{P}(\mathbf{y}|\mathbf{z})\,\mathbb{P}(\mathbf{z}, z^\star|\theta)}{\mathbb{P}(\mathbf{y}|\theta)},$$

where we have used $\mathbb{P}(\mathbf{y}|\mathbf{z}, z^\star) = \mathbb{P}(\mathbf{y}|\mathbf{z})$. Finally, the desired predictive distribution $\mathbb{P}(z^\star|\theta, \mathbf{y})$ is obtained by marginalizing out the latent variables in $\mathbf{z}$, *i.e.*,

$$\mathbb{P}(z^\star|\theta, \mathbf{y}) = \int \mathbb{P}(\mathbf{z}, z^\star|\theta, \mathbf{y}) dz = \frac{1}{\mathbb{P}(\mathbf{y}|\theta)} \int \mathbb{P}(\mathbf{y}|\mathbf{z})\,\mathbb{P}(\mathbf{z}, z^\star|\theta, \mathbf{y}) dz. \tag{1.5}$$

Since we have the joint Gaussian prior given in (1.4) and

$$\mathbf{y}|\mathbf{z} \sim \mathcal{N}\left(\mathbf{z}, {\sigma_\epsilon}^2 I\right),$$

the integral in (1.5) can be evaluated in closed-form and the predictive distribution turns out to be Gaussian, *i.e.*,

$$z^\star|\theta, \mathbf{y} \sim \mathcal{N}\left(\mu_{z^\star|\theta,\mathbf{y}}, \sigma^2_{z^\star|\theta,\mathbf{y}}\right), \tag{1.6}$$

where

$$\mu_{z^\star|\theta,\mathbf{y}} = \mu(x^\star) + k^T(\mathbf{K} + {\sigma_\epsilon}^2 I)^{-1}(\mathbf{y} - \mu), \tag{1.7}$$

and

$$\sigma^2_{z^\star|\theta,\mathbf{y}} = \mathcal{K}(x^\star, x^\star; \theta) - k^T(\mathbf{K} + {\sigma_\epsilon}^2 I)^{-1} k. \tag{1.8}$$

For notational simplicity, we define the covariance matrix of the noisy observations as $\mathbf{C} := \mathbb{C}\mathrm{ov}(\mathbf{y}, \mathbf{y}|\theta) = \mathbf{K} + {\sigma_\epsilon}^2 I$.

Chapter 2

Gaussian process regression under localization uncertainty

Abstract: In this chapter, we formulate Gaussian process regression with observations under the localization uncertainty due to the resource-constrained mobile sensor networks. In our formulation, effects of observations, measurement noise, localization uncertainty, and prior distributions are all correctly incorporated in the posterior predictive statistics. The analytically intractable posterior predictive statistics are proposed to be approximated by two techniques, viz., Monte Carlo sampling and Laplace's method. Such approximation techniques have been carefully tailored to our problems and their approximation error and complexity are analyzed. Simulation study demonstrates that the proposed approaches perform much better than approaches without considering the localization uncertainty properly. Finally, we have applied the proposed approaches on the experimentally collected real data from a dye concentration field over a section of a river and a temperature field of an outdoor swimming pool to provide proof of concept tests and evaluate the proposed schemes in real situations. In both simulation and experimental results, the proposed methods outperform the quick-and-dirty solutions often used in practice.

2.1 Introduction

Recently, there has been a growing interest in wireless sensor networks due to advanced embedded network technology. Their applications include, but are not limited to, environment monitoring, building comfort control, traffic control, manufacturing and plant automation, and surveillance systems [17, 25]. Mobility in a sensor network can increase its sensing coverage both in space and time, and robustness against uncertainties in environments. Exploitation of mobile sensor networks has been increased in collecting spatio-temporal data from the environment [10, 13, 64].

Gaussian process regression (or Kriging in geostatistics) has been widely used to draw statistical inference from geostatistical and environmental data [14, 84]. Gaussian process modeling enables us to predict physical values, such as temperature or harmful algae bloom biomass, at any point and time with a predicted uncertainty level. For example, near-optimal static sensor placements with a mutual information criterion in Gaussian processes were proposed in [56, 57]. A distributed Kriged Kalman filter for spatial estimation based on mobile sensor networks was developed in [13]. Centralized and distributed navigation strategies for mobile sensor networks to move in order to reduce prediction error variances at points of interest were developed in [118].

Localization in sensor networks is a fundamental problem in various applications to correctly fuse the spatially collected data and estimate the process of interest. However, obtaining precise localization of robotic networks under limited resources is very challenging [37,38]. Among the localization schemes, the range-based approach [50, 78] provides higher precision as compared to the range-free approach that could be cost-effective. The global positioning system (GPS) becomes one of the major absolute positioning systems in robotics and mobile sensor networks. Most of affordable GPSs slowly update their measurements and have resolution worse than one meter. A GPS is often augmented by the inertial navigation system (INS) for better resolution [102]. In practice, resource-constrained sensor network systems are prone to large uncertainty in localization. Most previous works on Gaussian process regression for mobile sensor networks [14, 57, 113, 118] have assumed that the exact sampling positions are available, which is not practical in real situations.

Therefore, motivated by the aforementioned issues, we consider correct (Bayesian) integration of uncertainties in sampling positions, and measurements noise for Gaussian process regression and its computation error and complexity analysis for the sensor network applications. The overall picture of our work is similar to the one in [74] in which an extended Kalman filter (EKF) was used to incorporate robot localization uncertainty and field parameter uncertainty. However, [74] relies on a parametric model, which is a radial basis function network model and EKF, while our motivation is to use more flexible non-parametric approach, viz., Gaussian process regression taking into account localization uncertainty in a Bayesian framework.

Gaussian process regression in [29] integrate uncertainty in the test input position for multiple-step ahead time series forecasting. In [29], uncertainty was not considered in the sampling positions of the training data (or observations). However, localization uncertainty effect is potentially significant. For example, Fig. 2.1 shows the effect of noisy sampling positions on the results of Gaussian process regression. Note that adding noise to the sampling positions considerably increase the empirical RMS error, shown in the third row of Fig. 2.1.

A Gaussian approximation to the intractable posterior predictive statistics obtained in [29] has been utilized for the predictive control with Gaussian models [54, 73] and Gaussian process dynamic programming [18]. In general, the length of the training data is much longer

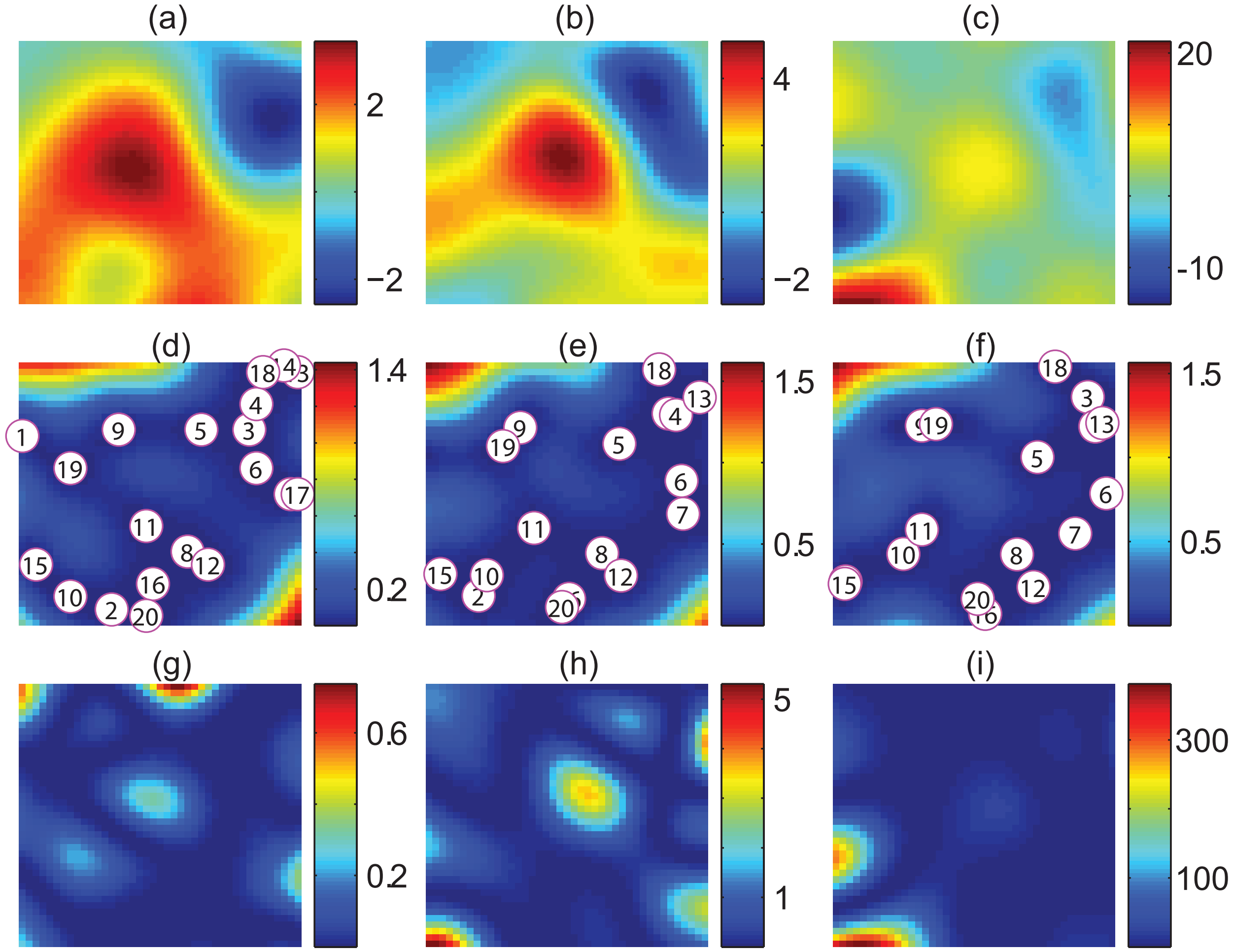


Figure 2.1: For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation. The prediction results of applying Gaussian process regression on the true and noisy sampling position are shown. The first, second, and third rows correspond to the prediction, prediction error variance, and squared empirical error (between predicted and true fields) fields. The first column shows the result of applying Gaussian process regression on the true sampling positions. Second and third columns show the results of applying Gaussian process regression on the noisy sampling positions with 0.1 and 0.4 noise covariance matrices, respectively. True and noisy sampling positions are shown in circles with agent indices in the second row.

than that of the test point for sensor network applications, therefore, our problem possibly involves a high dimensional uncertainty vector for the sampling positions.

Gaussian process prediction with localization uncertainty can be obtained as a posterior predictive distribution using Bayes' rule. The main difficulty to this is that it has no analytic closed-form solution and has to be approximated either through Monte Carlo sampling [75] or other approximation techniques such as variational inference [108]. As an important analytical approximation technique, Laplace's method has been known to be useful in many such situations [100, 101]. Different Laplace approximation techniques have been analyzed in terms of approximation error and computation complexity [68, 69, 100, 101].

The contribution of this chapter is as follows. First, we formulate Gaussian process regression with observations under the localization uncertainty due to the resource-constrained (possibly mobile) sensor networks. Next, approximations have been obtained for analytically intractable predictive mean, and predictive variance by using the Monte Carlo sampling and Laplace's method. Such approximation methods have been carefully tailored to our problems. In particular, a modified version of the moment generating function (MGF) approximation [101] has been developed as a part of Laplace's method to reduce the computational complexity. In addition, we have analyzed and compared the approximation error and the complexity so that one can choose a tradeoff between the performance requirements and constrained resources for a particular sensor network application. Another important contribution has been to provide proof of concept tests and evaluate the proposed schemes in real situations. We have applied the proposed approaches on the experimentally collected real data from a dye concentration field over a section of a river and a temperature field of an outdoor swimming pool.

The remainder of this chapter is organized as follows. In Section 2.2, we review Gaussian process regression, Monte Carlo sampling and Laplace's method briefly. In Section 2.3, Gaussian process prediction in the presence of the localization uncertainty has been formulated for a proposed sampling scheme as a Bayesian inference problem. We first present the Monte Carlo estimators for the posterior predictive mean and variance in Section 2.4. In

Section 2.5, using Laplace's method, we provide approximations for the posterior predictive statistics. Section 2.6 compares the computational cost and approximation accuracy over the proposed estimators based on the Mont Carlo sampling and the Laplace's method. Finally the simulation and experimental results are provided in Sections 2.7 and 2.8, respectively.

2.2 Preliminaries

In this section, we review the spatio-temporal Gaussian process, Monte Carlo sampling and Laplace's method, which will be used throughout the chapter.

2.2.1 Gaussian Process Regression

A Gaussian process defines a distribution over a space of functions and it is completely specified by its mean function and covariance function. Let $x \in \mathcal{X} := \mathcal{S} \times \mathcal{T} \subset \mathbb{R}^b$ denote the index vector, where $x := [\; s^T \quad t \;]^T$ contains the sampling position $s \in \mathcal{S} \subset \mathbb{R}^\iota$ and the sampling time $t \in \mathcal{T} \subset \mathbb{R}_{\geq 0}$. A Gaussian process, $z(x) \in \mathbb{R}$, is formally defined as below.

Definition 2.2.1. *A Gaussian process [84] is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

For notational simplicity, we consider a zero-mean Gaussian process[1] $z(x) \sim \mathcal{GP}(0, \mathcal{K}(x, x')) \in \mathbb{R}$. For example, one may consider a covariance function defined as

$$\mathcal{K}(x, x') = {\sigma_f}^2 \exp\left(-\frac{\left\|x - x'\right\|^2}{2\sigma_0^2}\right), \tag{2.1}$$

[1] A Gaussian process with a nonzero-mean can be treated by a change of variables. Even without a change of variables, this is not a drastic limitation, since the mean of the posterior process is not confined to zero [84].

where $x, x' \in \mathbb{R}^b$. In general, the mean and the covariance functions of a Gaussian process can be estimated a priori by maximizing the likelihood function [113].

Suppose, we have N noise corrupted observations without localization error, and $\mathcal{D}' = \{(x^{[i]}, y^{[i]}) \,|\, i = 1, \cdots, N\}$. Assume that $y^{[i]} = z^{[i]} + \epsilon^{[i]} \in \mathbb{R}$, where $\epsilon^{[i]}$ is an independent and identically distributed (i.i.d.) white Gaussian noise with variance ${\sigma_\epsilon}^2$. $\mathbf{x}$ is defined by $\mathbf{x} = \mathrm{col}(x^{[1]}, x^{[2]}, \cdots, x^{[N]}) \in \mathbb{R}^{bN}$. The collections of the realizations $\mathbf{z} = [\ z^{[1]} \ \ \ldots \ \ z^{[N]}\ ]^T \in \mathbb{R}^N$ and the observations $\mathbf{y} = [\ y^{[1]} \ \ \ldots \ \ y^{[N]}\ ]^T \in \mathbb{R}^N$ have the Gaussian distributions

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{K}(\mathbf{x})), \quad \mathbf{y} \sim \mathcal{N}(0, \mathbf{K}(\mathbf{x}) + {\sigma_\epsilon}^2 I),$$

where $\mathbf{K}(\mathbf{x}) \in \mathbb{R}^{N \times N}$ is the covariance matrix of $\mathbf{z}$ obtained by $(\mathbf{K}(\mathbf{x}))_{ij} = \mathcal{K}(x^{[i]}, x^{[j]})$, and I is the identity matrix with an appropriate size. We can predict the value $z^\star$ of the Gaussian process at a point $x^\star$ as [84]

$$z^\star | \mathcal{D}' \sim \mathcal{N}(\mu^\star(\mathbf{x}), \sigma^{\star 2}(\mathbf{x})). \tag{2.2}$$

In (2.2), the predictive mean is

$$\mu^\star(\mathbf{x}) = \mathbb{E}(z^\star | \mathcal{D}') = \mathbf{k}(\mathbf{x})^T (\mathbf{K}(\mathbf{x}) + {\sigma_\epsilon}^2 I)^{-1} \mathbf{y}, \tag{2.3}$$

and the predictive variance is given by

$$\sigma^{\star 2}(\mathbf{x}) = \mathbb{V}\mathrm{ar}(z^\star | \mathcal{D}') = {\sigma_f}^2 - k(\mathbf{x})^T (K(\mathbf{x}) + {\sigma_\epsilon}^2 I)^{-1} k(\mathbf{x}), \tag{2.4}$$

where $\mathbf{k}(\mathbf{x}) \in \mathbb{R}^N$ is the covariance matrix between $\mathbf{z}$ and $z^\star$ obtained by $\mathbf{k}_j(\mathbf{x}) = \mathcal{K}(x^{[j]}, x^\star)$, and ${\sigma_f}^2 = \mathcal{K}(x^\star, x^\star) \in \mathbb{R}$ is the variance at $x^\star$. (2.3) and (2.4) can be obtained from the fact that

$$\mathrm{col}(z^\star, \mathbf{y})|x^\star, \mathbf{x} \sim \mathcal{N}\left(0, \begin{bmatrix} {\sigma_f}^2 & \mathbf{k}(\mathbf{x})^T \\ \mathbf{k}(\mathbf{x}) & (\mathbf{K}(\mathbf{x}) + {\sigma_\epsilon}^2 I) \end{bmatrix}\right). \tag{2.5}$$

Note that the predictive mean in (2.3) and its prediction error variance in (2.4) require the inversion of the covariance matrix whose size depends on the number of observations N. Hence a drawback of Gaussian process regression is that its computational complexity and memory space increase as more measurements are collected, making the method prohibitive for agents with limited memory and computing power. To overcome this increase in complexity, a number of approximation methods for Gaussian process regression have been proposed. In particular, the sparse greedy approximation method [95], the Nystrom method [112], the informative vector machine [58], the likelihood approximation [93], and the Bayesian committee machine [104] have been shown to be effective for many problems.

In particular, it has been proposed that spatio-temporal Gaussian process regression can be applied to truncated observations including only measurements near the target position and time of interest for agents with limited resources [118]. To justify prediction based on only the most recent observations, a similar argument has been made in [6] in the sense that the data from the remote past do not change the predictors significantly under the exponentially decaying correlation functions.

In this chapter, we also assume that at each iteration the mobile sensor networks only needs to fuse a fixed number of truncated observations, which are near the target points of interest, to limit the computational resources.

2.2.2 Monte Carlo and Importance Sampling

In what follows, we briefly review Monte Carlo and Importance sampling based on [2]. The idea of Monte Carlo simulation is to draw an i.i.d. set of random samples $\{\mathbf{x}^{(i)}\}_{i=1}^{r}$ from a target density $\mathbb{P}(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}^N = \mathcal{R}$. These r random samples will be used to approximate $\mathbb{P}(\mathbf{x})$ by an empirical point-mass function $\mathbb{P}_r(\mathbf{x}) = \frac{1}{r}\sum\limits_{i=1}^{r} \delta(\mathbf{x}^{(i)} - \mathbf{x})$, where $\delta(\cdot)$ denotes the Dirac delta. Consequently, the integrals $I(f)$ can be approximated with tractable sums $I_r(f)$ that converge as follows.

$$I_r(f) = \frac{1}{r}\sum_{i=1}^{r} f(\mathbf{x}^{(i)}) \underset{r\to\infty}{\overset{a.s.}{\to}} I(f) = \int_{\mathcal{R}} f(\mathbf{x})\,\mathbb{P}(\mathbf{x})d\mathbf{x}.$$

$I_r(f)$ is an unbiased estimator. In addition, it will almost surely asymptotically converge to $I(f)$, which can be proved by the strong law of large numbers. Considering $f : \mathcal{R} \to \mathbb{R}$ for simplicity, if $f(\cdot)$ satisfies ${\sigma_e}^2 = \mathbb{E}(f^2(\mathbf{x})) - I^2(f) < \infty$, then the variance of the estimator $I_r(f)$ converges to $\frac{{\sigma_e}^2}{r}$ as r increases. In particular, the central limit theorem provides us with convergence in distribution as follows. $\sqrt{r}\,(I_r(f) - I(f)) \underset{r\to\infty}{\Rightarrow} \mathcal{N}(0, {\sigma_e}^2)$, where $\Rightarrow$ denotes convergence in distribution.

Importance sampling is a special case of Monte Carlo implementation, having random samples generated from an available distribution rather than the distribution of interest. Let us introduce an arbitrary importance proposal distribution $\rho(\mathbf{x})$ whose support includes the support of $\mathbb{P}(\mathbf{x})$. We can then rewrite $I(f)$ as $I(f) = \int_{\mathcal{R}} f(\mathbf{x})\omega(\mathbf{x})\rho(\mathbf{x})d\mathbf{x}$, where $\omega(\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x})}{\rho(\mathbf{x})}$ is known as the *importance weight*. Simulating r i.i.d. samples $\{\mathbf{x}^{(i)}\}_{i=1}^{r}$ according to $\rho(\mathbf{x})$ and evaluating $\omega(\mathbf{x}^{(i)})$, a possible unbiased Monte Carlo estimate of $I(f)$ is given by $\hat{I}_r(f) = \sum_{i=1}^{r} f(\mathbf{x}^{(i)})\omega(\mathbf{x}^{(i)}) / \sum_{i=1}^{r} \omega(\mathbf{x}^{(i)})$.

Under weak assumptions, the strong law of large numbers applies, yielding $\hat{I}_r(f) \underset{r\to\infty}{\overset{a.s.}{\to}} I(f)$.

This integration method can also be interpreted as a sampling method where the posterior density $\mathbb{P}(\mathbf{x})$ is approximated by $\hat{\mathbb{P}}_r(\mathbf{x}) = \sum_{i=1}^{r} \omega(\mathbf{x}^{(i)})\delta(\mathbf{x} - \mathbf{x}^{(i)}) / \sum_{i=1}^{r} \omega(\mathbf{x}^{(i)})$, and $\hat{I}_r(f)$ is the integration of $f(\mathbf{x})$ with respect to the empirical measure $\hat{\mathbb{P}}_r(\mathbf{x})$.

In this chapter, we shall compute the ratio of two integrals in the form of

$$G(f) = \frac{\int_{\mathcal{R}} f(\mathbf{x})\omega(\mathbf{x})\rho(\mathbf{x})d\mathbf{x}}{\int_{\mathcal{R}} \omega(\mathbf{x})\rho(\mathbf{x})d\mathbf{x}}, \tag{2.6}$$

where $\rho(\mathbf{x})$, $f(\mathbf{x})$ and $\omega(\mathbf{x})$ are defined on a space $\mathcal{R}$. To compute $G(f)$ in (2.6), we use the following approximation as proposed in [28].

$$\hat{G}_r(f) = \frac{\sum_{i=1}^{r} f(\mathbf{x}^{(i)})\omega(\mathbf{x}^{(i)})}{\sum_{i=1}^{r} \omega(\mathbf{x}^{(i)})}, \tag{2.7}$$

where $\{\mathbf{x}^{(i)} \,|\, i = 1, \cdots, r\}$ is a sequence of i.i.d. random vectors, which is drawn from $\rho(\mathbf{x})$ distribution.

In the following theorem, which can be shown to be a special case of the results from [28], we show the convergence properties of the approximation in (2.7) as functions of r, the number of random samples [28].

Theorem 2.2.2. *(Theorems 1 and 2 from [28]) Consider the approximation $\hat{G}_r(f)$ given in* (2.7) *to the ratio given $G(f)$ in* (2.6)*. If $\omega(\boldsymbol{x})\rho(\boldsymbol{x})$ is proportional to a proper probability density function defined on $\mathcal{R}$, and $G(f)$, $\int_{\mathcal{R}} \omega(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}$ and $\int_{\mathcal{R}} f^2(\boldsymbol{x})w^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}$ are finite, we have*

$$\hat{G}_r(f) \underset{r\to\infty}{\overset{a.s.}{\to}} G(f), \text{ and } \sqrt{r}\left(\hat{G}_r(f) - G(f)\right) \underset{r\to\infty}{\Rightarrow} \mathcal{N}(0, \sigma_g^2),$$

where

$$\sigma_g^2 = \int_{\mathcal{R}} (f(\boldsymbol{x}) - G(f))^2 w^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}.$$

Proof. It is a straightforward application of the results from Theorems 1 and 2 in [28] to (2.6) and (2.7). □

2.2.3 Laplace Approximations

The Laplace method is a family of asymptotic approximations that approximate an integral of a function, i.e., $\int_{\mathcal{R}} f(\mathbf{x})d\mathbf{x}$, where $\mathbf{x} \in \mathcal{R} \subset \mathbb{R}^m$, and $m = N \times b$. Let the function $f(\mathbf{x})$ be in a form $f(\mathbf{x}) = e^{-nh(\mathbf{x})}$, where $1 \ll n \in \mathbb{Z}_{>0}$, and $h : \mathcal{R} \to \mathbb{R}$ is a two times continuously differentiable real function on $\mathcal{R}$. Let $\hat{\mathbf{x}}$ denote the exact mode of $-h$, i.e.,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{R}} -h(\mathbf{x}).$$

Then Laplace's method produces the approximation [100]:

$$\int_{\mathcal{R}} f(\mathbf{x})d\mathbf{x} = \left(\frac{2\pi}{n}\right)^{\frac{m}{2}} |\phi(\hat{\mathbf{x}})|^{\frac{1}{2}} e^{-nh(\hat{\mathbf{x}})} + \mathcal{O}(n^{-1}), \tag{2.8}$$

where $\phi(\hat{\mathbf{x}}) = [\nabla^2 h(\hat{\mathbf{x}})]^{-1}$. The Laplace approximation in (2.8) will produce reasonable results as long as the $-h$ is unimodal or at least dominated by a single mode.

In practice it might be difficult to find the exact mode of $-h$. A concept of an asymptotic mode is introduced to gauge the approximation error when the exact mode is not used [68].

Definition 2.2.3. *$\breve{\boldsymbol{x}}$ is called an asymptotic mode of order $\mathcal{O}(n^{-k})$ for $-h$ if $\|\breve{\boldsymbol{x}} - \hat{\boldsymbol{x}}\| \to 0 \quad as \quad n \to \infty$, and $\nabla h(\breve{\boldsymbol{x}}) = \mathcal{O}(n^{-k})$.*

Suppose that $\breve{\mathbf{x}}$ is an asymptotic mode of order $\mathcal{O}(n^{-1})$ for $-h$ and $\{h, \breve{\mathbf{x}}\}$ satisfies the regularity conditions. Then it follows that we have

$$\int_{\mathcal{R}} f(\mathbf{x})d\mathbf{x} = \left(\frac{2\pi}{n}\right)^{\frac{m}{2}} |\phi(\breve{\mathbf{x}})|^{\frac{1}{2}} e^{-nh(\breve{\mathbf{x}})} C_h(\breve{\mathbf{x}}) + \mathcal{O}(n^{-1}), \tag{2.9}$$

where $C_h(\breve{\mathbf{x}})$ is given by

$$C_h(\breve{\mathbf{x}}) = e^{\frac{n}{2}\nabla h(\breve{\mathbf{x}})^T \phi(\breve{\mathbf{x}}) \nabla h(\breve{\mathbf{x}})}.$$

More precise form with the asymptotic mode of order $\mathcal{O}(n^{-2})$ is computed for an approximation of order $\mathcal{O}(n^{-3})$ in [69].

In many Bayesian inference applications and as in our problem, we need to compute the ratio of two integrals. To this end, *fully exponential Laplace approximations* has been developed by [100] to compute Laplace approximations of the ratio of two integrals, i.e.,

$$\mathcal{M} = \frac{\int_{\mathcal{R}} e^{-nu(\mathbf{x})} d\mathbf{x}}{\int_{\mathcal{R}} e^{-nh(\mathbf{x})} d\mathbf{x}}. \tag{2.10}$$

If each of $-u$ and $-h$ has a dominant peak at its maximum, then Laplace's method may be directly applied to both the numerator and denominator of (2.10) separately. If the regularity conditions are satisfied, using (2.8) for the denominator approximation and (2.9) for the numerator approximation, Miyata [68] obtained the following approximation and its error order ,

$$\begin{aligned} \hat{\mathcal{M}} &= \frac{|\nabla^2 h(\hat{\mathbf{x}})|^{\frac{1}{2}} e^{nh(\hat{\mathbf{x}})}}{|\nabla^2 u(\breve{\mathbf{x}})|^{\frac{1}{2}} e^{nu(\breve{\mathbf{x}})}} \times C_u(\breve{\mathbf{x}}), \\ \mathcal{M} &= \hat{\mathcal{M}} + \mathcal{O}(n^{-2}), \end{aligned} \tag{2.11}$$

where $\hat{\mathbf{x}}$ is the exact mode of $-h$, and $\breve{\mathbf{x}}$ is the asymptotic mode of $-u$, and

$$\begin{aligned} C_u(\breve{\mathbf{x}}) &= e^{\left(\frac{n}{2}\nabla u(\breve{\mathbf{x}})^T[\nabla^2 u(\breve{\mathbf{x}})]^{-1}\nabla u(\breve{\mathbf{x}})\right)}, \\ \breve{\mathbf{x}} &= \hat{\mathbf{x}} - [\nabla^2 u(\hat{\mathbf{x}})]^{-1}\nabla u(\hat{\mathbf{x}}). \end{aligned} \tag{2.12}$$

Laplace approximations typically has an error of order $\mathcal{O}(n^{-1})$ as shown in (2.8) and (2.9). On the other hand, fully exponential Laplace approximations for the ratio form yield an error of order $\mathcal{O}(n^{-2})$ as shown in (2.11) since the error terms of order $\mathcal{O}(n^{-1})$ in the numerator and denominator cancel each other [100].

Fully exponential Laplace approximations which are presented in (2.11) is limited for positive functions. Then, Tierney et al. [101] proposed a second-order approximation to the expectation of a general function $g(\mathbf{x})$ (not necessarily positive) by applying the fully exponential method to approximate $\mathcal{M}(\tau) = \mathbb{E}(e^{\tau g(\mathbf{x})})$ and then differentiating the approximated $\mathcal{M}(\tau)$. Consider $\mathcal{M}(\tau)$ is defined as follow

$$\mathcal{M}(\tau) = \frac{\int_{\mathcal{R}} e^{-nu(\mathbf{x})} d\mathbf{x}}{\int_{\mathcal{R}} e^{-nh(\mathbf{x})} d\mathbf{x}} = \frac{\int_{\mathcal{R}} e^{\tau g(\mathbf{x})} e^{-nh(\mathbf{x})} d\mathbf{x}}{\int_{\mathcal{R}} e^{-nh(\mathbf{x})} d\mathbf{x}},$$

where $u(\mathbf{x}) = -\frac{\tau}{n} g(\mathbf{x}) + h(\mathbf{x})$, and $e^{\tau g(\mathbf{x})}$ is positive, while $g(\mathbf{x})$ could be positive or negative. In particular, $\frac{d}{d\tau}\hat{\mathcal{M}}(\tau)$ evaluated at $\tau = 0$ yields a second-order approximation to $\mathbb{E}(g(\mathbf{x}))$ and its order of the error as follow.

$$\mathbb{E}(g(\mathbf{x})) = \frac{d}{d\tau}\mathcal{M}(\tau)\bigg|_{\tau=0} = \frac{d}{d\tau}\hat{\mathcal{M}}(\tau)\bigg|_{\tau=0} + \mathcal{O}(n^{-2}). \tag{2.13}$$

This method, which is called moment generating function (MGF) Laplace approximation, gives a standard-form approximation using the exact mode of $-h(\mathbf{x})$ [101].

Miyata [68,69] extended the MGF method for one without computing the exact mode of $-h(\mathbf{x})$. Let $\hat{\mathbf{x}}$ be an asymptotic mode of order $\mathcal{O}(n^{-1})$ for $-h(\mathbf{x})$. Suppose that $\{h, \hat{\mathbf{x}}\}$ satisfies the regularity conditions for the asymptotic-mode Laplace method. By using Theorem 5 in [68], the approximation of $\hat{\mathbb{E}}(g(\mathbf{x}))$ and its error order are given as

$$\begin{aligned}\hat{\mathbb{E}}(g(\mathbf{x})) &= g(\hat{\mathbf{x}}) + \frac{1}{2n}\operatorname{tr}\left(\nabla^2 g(\hat{\mathbf{x}})\phi(\hat{\mathbf{x}})\right) - \frac{1}{2n}\sum_{ijk\ell=1}^{m}\frac{\partial^3 h(\hat{\mathbf{x}})}{\partial \mathbf{x}_i \partial \mathbf{x}_j \partial \mathbf{x}_k}\phi_{i\ell}\phi_{jk}\frac{\partial g(\hat{\mathbf{x}})}{\partial \mathbf{x}_\ell} \\ &\quad - \nabla g(\hat{\mathbf{x}})^T \phi(\hat{\mathbf{x}}) \nabla h(\hat{\mathbf{x}}), \\ \mathbb{E}(g(\mathbf{x})) &= \hat{\mathbb{E}}(g(\mathbf{x})) + \mathcal{O}(n^{-2}),\end{aligned} \tag{2.14}$$

where ϕ_{ij} is the i-th row, j-th column element of the matrix $\phi(\hat{\mathbf{x}})$. Furthermore, if $\hat{\mathbf{x}}$ is the exact mode of $-h(\mathbf{x})$, then the approximation has a simpler form because the terms that include $\nabla h(\hat{\mathbf{x}})$ vanish

$$\hat{\mathbb{E}}(g(\mathbf{x})) = g(\hat{\mathbf{x}}) + \frac{1}{2n}\operatorname{tr}\left(\nabla^2 g(\hat{\mathbf{x}})\phi(\hat{\mathbf{x}})\right) - \frac{1}{2n}\sum_{ijk\ell=1}^{m}\frac{\partial^3 h(\hat{\mathbf{x}})}{\partial \mathbf{x}_i \partial \mathbf{x}_j \partial \mathbf{x}_k}\phi_{i\ell}\phi_{jk}\frac{\partial}{\partial \mathbf{x}_\ell} g(\hat{\mathbf{x}}). \tag{2.15}$$

2.3 The Problem Statement

In practice, $\mathcal{D}'$ (data with perfect localization) is not available due to localization uncertainty, and instead its exact sampling points will be replaced with noise corrupted sampling points. To average out measurement and localization noises, in this chapter, we propose to use a sampling scheme in which multiple measurements are taken repeatedly at a set of sampling points of a sensor network. For robotic sensors or mobile sensor networks, this sampling strategy could be efficient and inexpensive since the large energy consumption is usually due to the mobility of the sensor network. Let N sensing agents be indexed by $\mathcal{J} = \{1, \cdots, N\}$.

From the proposed sampling scheme, we assume that each agent takes multiple data pairs $\{(\bar{x}^{[i]}, y^{[i]}) \,|\, i \in \mathcal{I}\}$, which are indexed by $\mathcal{I} = \{1, \cdots, n\}$ at a set of sampling points by the sensor network $\{x^{[j]} \,|\, j \in \mathcal{J}\}$. We then define the map $\psi : \mathcal{I} \to \mathcal{J}$ that takes the index of the data pair in $\mathcal{I}$ as the input and returns the index of the sensor that produced the data pair as the output. Consider the following realizations using the sampling scheme and the notation just introduced.

$$\begin{aligned} \bar{x}^{[i]} &= x^{(\psi(i))} + v^{[i]} \in \mathbb{R}^b, \quad \forall i \in \mathcal{I} \\ y^{[i]} &= z^{(\psi(i))} + \epsilon^{[i]} \in \mathbb{R}, \quad \forall i \in \mathcal{I}, \end{aligned}$$

where $\epsilon^{[i]}$ is an i.i.d. white Gaussian noise with a zero mean and a variance of ${\sigma_\epsilon}^2$, i.e., $\epsilon^{[i]} \sim \mathcal{N}(0, {\sigma_\epsilon}^2)$ and $v^{[i]}$ is a localization error which has a multivariate normal distribution with a zero mean and a covariance matrix $\Sigma_v \in \mathbb{R}^{b \times b}$, i.e., $v^{[i]} \sim \mathcal{N}(0, \Sigma_v)$. For instance, the distribution of the localization error may be a result of the fusion of GPS and INS measurements [102], or landmark observations and robot's kinematics [24].

To simplify the notation, $\mathcal{D}$ is introduced to denote the data with the measurement noise and localization error as follows.

$$\mathcal{D} = \left\{ (\bar{x}^{[i]}, y^{[i]}) \,|\, i \in \mathcal{I} \right\}. \tag{2.16}$$

We also define the collective sampling point vectors with and without localization uncertainty, and the cumulative localization noise vector, respectively by

$$\begin{aligned}
\mathbf{x} &= \mathrm{col}(x^{[1]}, x^{[2]}, \cdots, x^{[N]}) \in \mathbb{R}^{bN}, \\
\bar{\mathbf{x}} &= \mathrm{col}(\bar{x}^{[1]}, \bar{x}^{[2]}, \cdots, \bar{x}^{(n)}) \in \mathbb{R}^{bn}, \\
\mathbf{v} &= \mathrm{col}(v^{[1]}, v^{[2]}, \cdots, v^{(n)}) \in \mathbb{R}^{bn}.
\end{aligned} \tag{2.17}$$

From the proposed sampling scheme, to average out the measurement and localization uncertainties, the number of measurements n can be increased without increasing the number of sensors N, and consequently without increasing the dimension of $\mathbf{x} \in \mathbb{R}^{bN}$. Hence, this approach may be efficient for the resource-constrained (mobile) sensor network at the cost of taking more measurements. Using collective sampling point vectors in (2.17), we have the following relationship.

$$\bar{\mathbf{x}} = L\mathbf{x} + \mathbf{v}, \tag{2.18}$$

where $L = \bar{L} \otimes I_b \in \mathbb{R}^{bn \times bN}$, $\bar{L} \in \mathbb{R}^{n \times N}$ and $\big(\bar{L}\big)_{ij} = 1$ if $\psi(i) = j$, otherwise $\big(\bar{L}\big)_{ij} = 0$.

The conditional probabilities $\mathbb{P}(\bar{\mathbf{x}}|\mathbf{x})$ and $\mathbb{P}(\mathbf{x}|\bar{\mathbf{x}})$ can be expressed as

$$\begin{aligned}
\mathbb{P}(\bar{\mathbf{x}}|\mathbf{x}) =& \frac{1}{|2\pi\Sigma_{\mathbf{v}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\bar{\mathbf{x}}-L\mathbf{x})^T \Sigma_{\mathbf{v}}^{-1} (\bar{\mathbf{x}}-L\mathbf{x})}, \\
\mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) =& \frac{\mathbb{P}(\bar{\mathbf{x}}|\mathbf{x})\, \mathbb{P}(\mathbf{x})}{\int_{\mathcal{R}} \mathbb{P}(\bar{\mathbf{x}}|\mathbf{x})\, \mathbb{P}(\mathbf{x}) d\mathbf{x}},
\end{aligned}$$

where $\Sigma_{\mathbf{v}} = I_n \otimes \Sigma_v \in \mathbb{R}^{bn \times bn}$. From a Bayesian perspective, we can treat $\mathbf{x}$ as a random vector to incorporate a prior distribution on $\mathbf{x}$. For example, if we assign a prior distribution on $\mathbf{x}$ such that $\mathbf{x} \sim \mathcal{N}(0, \Sigma_{\mathbf{x}})$ then we have

$$\mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) = \frac{1}{|2\pi\mathring{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-H\bar{\mathbf{x}})^T \mathring{\Sigma}^{-1} (\mathbf{x}-H\bar{\mathbf{x}})}. \tag{2.19}$$

where $H = \mathring{\Sigma} L^T \Sigma_{\mathbf{v}}^{-1}$ and $\mathring{\Sigma}^{-1} = \Sigma_{\mathbf{x}}^{-1} + L^T \Sigma_{\mathbf{v}}^{-1} L$.

Evaluating posterior predictive statistics such as the density, the mean, and the variance are of critical importance for the sensor network applications.

Therefore, given the data $\mathcal{D}$ in (2.16), our goal is to compute the posterior predictive statistics. In particular we focused on the following two quantities given in detail.

- The predictive mean estimator (PME) is given by $\mathbb{E}(z^\star|\mathcal{D}) = \int_{\mathcal{Z}} z^\star \, \mathbb{P}(z^\star|\mathcal{D}) dz^\star$.

$$\begin{aligned} \mathbb{E}(z^\star|\mathcal{D}) &= \frac{\int_{\mathcal{Z}} z^\star \int_{\mathcal{R}} \mathbb{P}(z^\star|\mathbf{x},\mathbf{y}) \, \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x} dz^\star}{\int_{\mathcal{R}} \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}} \\ &= \frac{\int_{\mathcal{R}} \mu^\star(\mathbf{x}) \, \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}}{\int_{\mathcal{R}} \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}}, \end{aligned} \tag{2.20}$$

  where $\mu^\star(\mathbf{x})$ is given by (2.3).

- The predictive variance estimator (PVE) is obtained similarly. From the following equation,

$$\mathbb{E}(z^{\star 2}|\mathcal{D}) = \frac{\int_{\mathcal{R}} \left(\int_{\mathcal{Z}} z^{\star 2} \, \mathbb{P}(z^\star|\mathbf{x},\mathbf{y}) dz^\star \right) \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}}{\int_{\mathcal{R}} \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}},$$

  where $\int_{\mathcal{Z}} z^{\star 2} \, \mathbb{P}(z^\star|\mathbf{x},\mathbf{y}) dz^\star = \sigma^{\star 2}(\mathbf{x}) + \mu^{\star 2}(\mathbf{x})$, we obtain $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D})$ given as the following formula.

$$\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D}) = \frac{\int_{\mathcal{R}} (\sigma^{\star 2}(\mathbf{x}) + \mu^{\star 2}(\mathbf{x})) \, \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}}{\int_{\mathcal{R}} \mathbb{P}(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}} - \mathbb{E}^2(z^\star|\mathcal{D}), \tag{2.21}$$

  where $\sigma^{\star 2}(\mathbf{x})$ is given by (2.4).

The main challenge to our problems is the fact that there are no closed-form formulas for the posterior predictive statistics listed in (2.20), and (2.21). Therefore, in this chapter, approximation techniques will be carefully applied to obtain approximate solutions. In addition, the tradeoffs between the computational complexity and the precision will be investigated for the sensor networks with limited resources.

From (2.19), one might be tempted to use the best estimate of $\mathbf{x}$, e.g., the conditional expectation of $\mathbf{x}$ for given measured locations $\bar{\mathbf{x}}$, i.e., $\mathbb{E}(\mathbf{x}|\bar{\mathbf{x}})$ for Gaussian process regression. Comparison between these type of quick-and-dirty solutions and the proposed Bayesian approaches will be evaluated and discussed with simulated and experimental data presented in Sections 2.7 and 2.8, respectively.

2.4 The Monte Carlo method

In this section, we propose importance sampling to compute the posterior predictive statistics. We also summarize the convergence results of the MC estimators based on Theorem 2.2.2.

2.4.1 Predictive mean

The predictive mean estimator is given by (2.20). Using importance sampling to approximate (2.20) leads to the following equation,

$$\hat{\mathbb{E}}(z^\star|\mathcal{D}) = \frac{\sum_{i=1}^{r} \mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})\mu^\star(\mathbf{x}^{(i)})}{\sum_{i=1}^{r} \mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})}. \tag{2.22}$$

Theorem 2.2.2 and (2.22) lead to

$$\hat{\mathbb{E}}(z^\star|\mathcal{D}) \underset{r\to\infty}{\overset{a.s.}{\to}} \mathbb{E}(z^\star|\mathcal{D}),$$
$$\left(\mathbb{E}(z^\star|\mathcal{D}) - \hat{\mathbb{E}}(z^\star|\mathcal{D})\right) \underset{r\to\infty}{\Rightarrow} \mathcal{N}\left(0, \frac{\sigma_m^2}{r}\right),$$

where $\sigma_m^2 = \int_{\mathcal{R}} (\mu^\star(\mathbf{x}) - \mathbb{E}(z^\star|\mathcal{D}))^2 \, \mathbb{P}^2(\mathbf{y}|\mathbf{x}) \, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}$, $\mu^\star(\mathbf{x})$ is calculated from (2.3), and $\mathbf{x}^{(i)}$ has been sampled from $\mathbb{P}(\mathbf{x}|\bar{\mathbf{x}})$ given by (2.19).

2.4.2 Predictive variance

For the prediction error variance given by (2.21), we have

$$\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D}) = \mathbb{E}\left(\sigma^{\star 2}(\mathbf{x})\right) + \mathbb{V}\mathrm{ar}\left(\mu^\star(\mathbf{x})\right).$$

Thus, we propose the following estimator.

$$\begin{aligned}\widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) =& \frac{\sum_{i=1}^r \mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})\sigma^{\star 2}(\mathbf{x}^{(i)})}{\sum_{i=1}^r \mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})} \\ &+ \frac{\sum_{i=1}^r \mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})(\mu^\star(\mathbf{x}^{(i)}) - \hat{\mathbb{E}}(z^\star|\mathcal{D}))^2}{\sum_{i=1}^r \mathbb{P}(\mathbf{y}|\mathbf{x}^{(i)})},\end{aligned} \tag{2.23}$$

where $\sigma^{\star 2}(\mathbf{x}^{(i)}) = \mathbb{V}\mathrm{ar}(z^\star|\mathbf{x}^{(i)}, \mathbf{y})$, $\mu^\star(\mathbf{x}^{(i)}) = \mathbb{E}(z^\star|\mathbf{x}^{(i)}, \mathbf{y})$ are obtained from the formulas similar to (2.4) and (2.3), and $\hat{\mathbb{E}}(z^\star|\mathcal{D})$ is given by (2.22).

Applying Theorem 2.2.2 for $f(x) = \sigma^{\star 2}(\mathbf{x}) + (\mu^\star(\mathbf{x}) - \hat{\mathbb{E}}(z^\star|\mathcal{D}))^2$, $\omega(\mathbf{x}) = \mathbb{P}(\mathbf{y}|\mathbf{x})$, and $\rho(\mathbf{x}) = \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}})$, we obtain the following results.

$$\widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) \underset{r\to\infty}{\overset{a.s.}{\to}} \overline{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}),$$

where

$$\overline{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) = \mathbb{E}(z^{\star 2}|\mathcal{D}) - 2\mathbb{E}(z^\star|\mathcal{D})\hat{\mathbb{E}}(z^\star|\mathcal{D}) + \hat{\mathbb{E}}^2(z^\star|\mathcal{D}),$$

$$\left(\overline{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) - \widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D})\right) \underset{r\to\infty}{\Rightarrow} \mathcal{N}\left(0, \sigma_z^2 r^{-1}\right),$$

and

$$\sigma_z^2 = \int_{\mathcal{R}} (\sigma^{\star 2}(\mathbf{x}) + (\mu^\star(\mathbf{x}) - \hat{\mathbb{E}}(z^\star|\mathcal{D}))^2 - \overline{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}))^2\, \mathbb{P}^2(\mathbf{y}|\mathbf{x})\, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}) d\mathbf{x}.$$

Remark 2.4.1. *Since $\mathbb{E}(z^\star|\mathcal{D})$ is not available in (2.23), $\hat{\mathbb{E}}(z^\star|\mathcal{D})$ is used instead. Subsequently, the convergence results for $\widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D})$ are given with respect to $\overline{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D})$ which converges to $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D})$ as $r \to \infty$ by the definition.*

2.5 Fully Exponential Laplace Approximations

In this section, we propose fully exponential Laplace approximations to compute the posterior predictive statistics. In the process of applying Laplace approximations, we also obtain the estimation of the sampling points given $\mathcal{D}$ as a by-product. From the observations $\mathcal{D}$ given by (2.16), we can update the estimates of the sampling points $\mathbf{x}$. To this end, we use the maximum a posteriori probability (MAP) estimate of $\mathbf{x}$ given by

$$\hat{\mathbf{x}}_{MAP} = \arg\max_{\mathbf{x}\in\mathcal{R}} \mathbb{P}(\mathbf{y}|\mathbf{x})\, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}). \tag{2.24}$$

2.5.1 Perdictive mean

The predictive mean estimator, given by (2.20), can be approximated using MGF method (2.14). To compute $\mathbb{E}(z^\star|\mathcal{D})$, let $g(\mathbf{x}) = \mu^\star(\mathbf{x})$. Using

$$h(\mathbf{x}) = -\frac{1}{n} \ln \left(\mathbb{P}(\mathbf{y}|\mathbf{x})\, \mathbb{P}(\mathbf{x}|\bar{\mathbf{x}})\right), \tag{2.25}$$

the predictive mean estimation using MGF method (MGF-PME), $\hat{\mathbb{E}}(z^\star|\mathcal{D})$ and its error order are given as

$$\begin{aligned} &\hat{\mathbb{E}}(z^\star|\mathcal{D}) = \hat{\mathbb{E}}(g(\mathbf{x})), \\ &\mathbb{E}(z^\star|\mathcal{D}) = \hat{\mathbb{E}}(z^\star|\mathcal{D}) + \mathcal{O}(n^{-2}), \end{aligned} \tag{2.26}$$

where $\hat{\mathbb{E}}(g(\mathbf{x}))$ is given in (2.14) or (2.15). However, the MGF-PME given by (2.26) needs the computation of the third derivative of h, which increases the complexity of the algorithm.

In this chapter, another MGF method has been developed in order not to use the third derivative of h. To approximate the derivative of $\mathcal{M}(\cdot)$ at a point τ in (2.13), we utilize a three-point estimation, which is the slope of a nearby secant line through the points $(\tau - \zeta, \mathcal{M}(\tau - \zeta))$ and $(\tau + \zeta, \mathcal{M}(\tau + \zeta))$. Approximating the derivative in (2.13) with the three-point estimation, we can avoid the third derivative in (2.14) or (2.15). We summarize our results in the following theorem.

Theorem 2.5.1. *Let $\hat{\boldsymbol{x}}$ be the exact mode of $-h(\boldsymbol{x})$. The three-point predictive mean esti-*

mator (TP-PME) and its order of the error are given by

$$\begin{aligned}
\hat{\mathbb{E}}(z^\star|\mathcal{D}) &= \frac{1}{2} n^{\frac{3}{4}} \left|\nabla^2 h(\hat{\boldsymbol{x}})\right|^{\frac{1}{2}} e^{nh(\hat{\boldsymbol{x}})} \\
&\quad \times \left\{ \left|\nabla^2 \mathring{b}(\hat{\boldsymbol{x}}_{\mathring{b}})\right|^{-\frac{1}{2}} C_{\mathring{b}}(\hat{\boldsymbol{x}}_{\mathring{b}}) e^{-n\mathring{b}(\hat{\boldsymbol{x}}_{\mathring{b}})} - \left|\nabla^2 \mathring{p}(\hat{\boldsymbol{x}}_{\mathring{p}})\right|^{-\frac{1}{2}} C_{\mathring{p}}(\hat{\boldsymbol{x}}_{\mathring{p}}) e^{-n\mathring{p}(\hat{\boldsymbol{x}}_{\mathring{p}})} \right\}, \qquad (2.27) \\
\hat{\mathbb{E}}(z^\star|\mathcal{D}) &= \mathbb{E}(z^\star|\mathcal{D}) + \mathcal{O}(n^{-3/2}),
\end{aligned}$$

where $h(\boldsymbol{x})$ *is given by* (2.25)*, and we have used the following definitions*

$$\begin{aligned}
\mathring{b}(\boldsymbol{x}) &= h(\boldsymbol{x}) - n^{-\frac{7}{4}} \mu^\star(\boldsymbol{x}), \\
\mathring{p}(\boldsymbol{x}) &= h(\boldsymbol{x}) + n^{-\frac{7}{4}} \mu^\star(\boldsymbol{x}), \\
\hat{\boldsymbol{x}}_{\mathring{b}} &= \hat{\boldsymbol{x}} - [\nabla^2 \mathring{b}(\hat{\boldsymbol{x}})]^{-1} \nabla \mathring{b}(\hat{\boldsymbol{x}}), \\
\hat{\boldsymbol{x}}_{\mathring{p}} &= \hat{\boldsymbol{x}} - [\nabla^2 \mathring{p}(\hat{\boldsymbol{x}})]^{-1} \nabla \mathring{p}(\hat{\boldsymbol{x}}), \\
C_{\mathring{b}}(\hat{\boldsymbol{x}}_{\mathring{b}}) &= e^{\frac{n}{2} \nabla \mathring{b}(\hat{\boldsymbol{x}}_{\mathring{b}})^T [\nabla^2 \mathring{b}(\hat{\boldsymbol{x}}_{\mathring{b}})]^{-1} \nabla \mathring{b}(\hat{\boldsymbol{x}}_{\mathring{b}})}, \\
C_{\mathring{p}}(\hat{\boldsymbol{x}}_{\mathring{p}}) &= e^{\frac{n}{2} \nabla \mathring{p}(\hat{\boldsymbol{x}}_{\mathring{p}})^T [\nabla^2 \mathring{p}(\hat{\boldsymbol{x}}_{\mathring{p}})]^{-1} \nabla \mathring{p}(\hat{\boldsymbol{x}}_{\mathring{p}})}.
\end{aligned}$$

Proof. First we use the three-point method to approximate derivative such that

$$\left.\frac{d}{d\tau}\mathcal{M}(\tau)\right|_{\tau=0} = \frac{\mathcal{M}(\zeta) - \mathcal{M}(-\zeta)}{2\zeta} + \mathcal{O}(\zeta^2).$$

Plugging $\mathcal{M}(\zeta) = \hat{\mathcal{M}}(\zeta) + \mathcal{O}(n^{-2})$ into the above equation and using (2.13), we obtain

$$\hat{\mathbb{E}}(z^\star|\mathcal{D}) = \frac{\hat{\mathcal{M}}(\zeta) - \hat{\mathcal{M}}(-\zeta)}{2\zeta} + \mathcal{O}(n^{-2})\,\mathcal{O}(\zeta^{-1}) + \mathcal{O}(\zeta^2). \qquad (2.28)$$

By selecting $\zeta = n^{-3/4}$, we recover the order of the error $\mathcal{O}(n^{-2})\,\mathcal{O}(\zeta^{-1}) + \mathcal{O}(\zeta^2) = \mathcal{O}(n^{-3/2})$. By computing the estimates $\hat{\mathcal{M}}(\zeta)$ and $\hat{\mathcal{M}}(-\zeta)$ in (2.28) using (2.11), (2.27) is obtained. □

Remark 2.5.2. *The complexity of either* (2.14) *or* (2.15) *is* $\mathcal{O}(n^4)$ *while the complexity of* (2.27) *is* $\mathcal{O}(n^3)$*. In return, the error of* (2.15) *is of order* $\mathcal{O}(n^{-2})$ *and the error of* (2.27) *is of order* $\mathcal{O}(n^{-3/2})$*.*

2.5.2 Predictive variance

We now apply Laplace's method to approximate the prediction error variance in a similar way. The prediction error variance is given by (2.21). In this case, $h(\mathbf{x})$ is given by (2.25) and $u(\mathbf{x}) = -\frac{1}{n}\left(\ln(\sigma^{\star 2}(\mathbf{x}) + \mu^{\star 2}(\mathbf{x})) + \ln(\mathbb{P}(\mathbf{y}|\mathbf{x})\,\mathbb{P}(\mathbf{x}|\bar{\mathbf{x}}))\right)$. Applying (2.11) to this case, the approximate of $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D})$ and its order of the error are given by

$$\begin{aligned}\widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) =& \frac{|\nabla^2 h(\hat{\mathbf{x}})|^{\frac{1}{2}} e^{nh(\hat{\mathbf{x}})}}{|\nabla^2 u(\breve{\mathbf{x}})|^{\frac{1}{2}} e^{nu(\breve{\mathbf{x}})}} C_u(\breve{\mathbf{x}}) - \hat{\mathbb{E}}^2(z^\star|\mathcal{D}),\\ \widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) =& \mathbb{V}\mathrm{ar}(z^\star|\mathcal{D}) + \mathcal{O}(n^{-3/2}),\end{aligned} \tag{2.29}$$

where $\hat{\mathbf{x}}$ is the exact mode of $-h$, and $\breve{\mathbf{x}}$ is the asymptotic mode of $-u$. $C_u(\breve{\mathbf{x}})$ and $\breve{\mathbf{x}}$ are given by (2.12) and $\hat{\mathbb{E}}(z^\star|\mathcal{D})$ is given by (2.27).

2.5.3 Simple Laplace approximations

To minimize the computational complexity, one may prefer a simpler approximation. In this chapter, we propose such a simple approximation at the cost of precision, which is summarized in the following theorem.

Theorem 2.5.3. *Let* $\hat{\boldsymbol{x}}$ *be an asymptotic mode of order* $\mathcal{O}(n^{-1})$ *for* $-h$ *given by* (2.25)*. Assume that* $\{h, \hat{\boldsymbol{x}}\}$ *satisfies the regularity conditions. Consider the following approximations* *for* $\mathbb{E}(z^\star|\mathcal{D})$ *and* $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D})$

$$\hat{\mathbb{E}}(z^\star|\mathcal{D}) = \boldsymbol{k}^T(\hat{\boldsymbol{x}})(\boldsymbol{K}(\hat{\boldsymbol{x}}) + {\sigma_\epsilon}^2 I)^{-1}\boldsymbol{y}, \tag{2.30}$$

$$\widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) = {\sigma_f}^2 - \boldsymbol{k}^T(\hat{\boldsymbol{x}})(\boldsymbol{K}(\hat{\boldsymbol{x}}) + {\sigma_\epsilon}^2 I)^{-1}\boldsymbol{k}(\hat{\boldsymbol{x}}), \tag{2.31}$$

where $\boldsymbol{K}(\hat{\boldsymbol{x}})$ *and* $\boldsymbol{k}(\hat{\boldsymbol{x}})$ *are covariance matrices as in* (2.3) *but obtained with* $\hat{\boldsymbol{x}}$*.*

We have then the following order of errors.

$$\hat{\mathbb{E}}(z^\star|\mathcal{D}) = \mathbb{E}(z^\star|\mathcal{D}) + \mathcal{O}(n^{-1}),$$
$$\widehat{\mathbb{V}\mathrm{ar}}(z^\star|\mathcal{D}) = \mathbb{V}\mathrm{ar}(z^\star|\mathcal{D}) + \mathcal{O}(n^{-1}).$$

Proof. The approximation for $\mathbb{E}(z^\star|\mathcal{D})$ given by (2.30) is the first term of (2.14) neglecting high order terms. The second and third terms in (2.14) have n^{-1} inside. $\hat{\mathbf{x}}$ is an asymptotic mode of order $\mathcal{O}(n^{-1})$ for $-h$. By Definition 2.2.3, the last term of (2.14) that contains $\nabla h(\hat{\mathbf{x}})$ is $\mathcal{O}(n^{-1})$. Hence, without these high order terms, we obtain a simpler approximation of order $\mathcal{O}(n^{-1})$.

The approximation for $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D})$ given by (2.31) can be obtained by approximating (2.21) with the first term of the MGF approximation. Let $g(\mathbf{x}) = \sigma^{\star 2}(\mathbf{x}) + \mu^{\star 2}(\mathbf{x})$, then $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D}) = \mathbb{E}(g(\mathbf{x})|\mathcal{D}) - \mathbb{E}^2(z^\star|\mathcal{D})$. Using the first term of MGF $\mathbb{E}(g(\mathbf{x})|\mathcal{D}) = g(\hat{\mathbf{x}}) + \mathcal{O}(n^{-1}) = \sigma^{\star 2}(\hat{\mathbf{x}}) + \mu^{\star 2}(\hat{\mathbf{x}}) + \mathcal{O}(n^{-1})$, and using $\mathbb{E}^2(z^\star|\mathcal{D}) = \mu^{\star 2}(\hat{\mathbf{x}}) + \mathcal{O}(n^{-1})$, we obtain $\mathbb{V}\mathrm{ar}(z^\star|\mathcal{D}) = \sigma^{\star 2}(\hat{\mathbf{x}}) + \mathcal{O}(n^{-1})$. The procedure of showing the order of the approximation error is the same as what was shown for the approximation in (2.30). □

Remark 2.5.4. *Note that the simple Laplace predictive mean and variance estimators in* (2.30) *and* (2.31) *take the same forms of the original predictive mean and variance without* *the localization error, respectively given in* (2.3) *and* (2.4)*, evaluated at the MAP estimator of* $\boldsymbol{x}$*. As we previously mentioned,* $\hat{\boldsymbol{x}}$ *is the mode of* $-h$ *given by* (2.25) *and is the MAP estimator of* $\boldsymbol{x}$*, i.e.,* $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{MAP}$ *as defined in* (2.24)*. Therefore, the difference in the simple Laplace approximations with respect to a quick-and-dirty solution in which the measured location vector* $\bar{\boldsymbol{x}}$ *is used is that the simple Laplace approximations use* $\hat{\boldsymbol{x}}$ *instead of* $\bar{\boldsymbol{x}}$*.*

In applying Laplace's method, using the one step Newton gradient method to compute asymptotic modes, e.g., $\check{\mathbf{x}}$ required in (2.29) or $\hat{\mathbf{x}}_{\mathring{b}}$ and $\hat{\mathbf{x}}_{\mathring{p}}$ required in (2.27) may not satisfy the regularity conditions. In this case, one needs to continue the Newton gradient optimization until the regularity conditions are satisfied.

2.6 Error and Complexity Analysis

In this section, we analyze the order of the error and the computational complexity for the proposed approximation methods, which are summarized in Table 2.1. A tradeoff between the approximation error and complexity can be chosen taking into account the performance requirements and constrained resources for a specific sensor network application.

For Laplace's method, the order of the error ranges from $\mathcal{O}(n^{-1})$ to $\mathcal{O}(n^{-2})$ at the cost of complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^4)$ as shown in Table 2.1.

For the Monte Carlo estimators, we introduce $\mathbb{O}(\sigma_r)$, which is the probabilistic error order and it implies that the estimation error converges to $\mathcal{N}(0, \sigma_r^2)$ in distribution as the number of random samples r increases. Therefore it is not appropriate to compare the error bound between Monte Carlo and Laplace's method exactly. Monte Carlo algorithms are used for multivariate integration of dimension $b \times N$. The probabilistic error order of Monte Carlo algorithms that use r sample evaluations is of order $r^{-1/2}$ for a given n. We may

Table 2.1: Error and complexity analysis

Estimator	Method	Error	Complexity
MC-PME in (2.22)	Monte Carlo	$\mathbb{O}\left(\frac{\sigma_m}{\sqrt{r}}\right)$	$\mathcal{O}(n^3 \times r)$
MGF-PME in (2.14)	Laplace MGF	$\mathcal{O}(n^{-2})$	$\mathcal{O}(n^4)$
TP-PME in (2.27)	Laplace MGF	$\mathcal{O}(n^{-3/2})$	$\mathcal{O}(n^3)$
S-PME in (2.30)	Laplace MGF	$\mathcal{O}(n^{-1})$	$\mathcal{O}(n^3)$

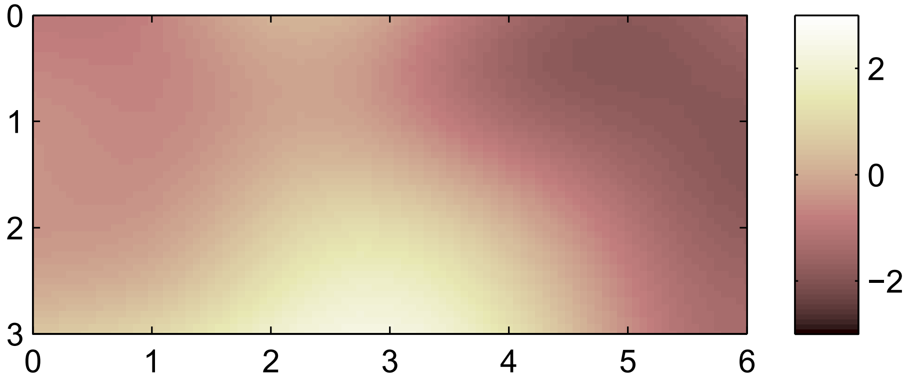


Figure 2.2: A realization of the Gaussian process-ground truth.

assume that the order of the error for Monte Carlo methods do not depend on the number of measurements n for a large n [28]. With this assumption, the number of samples needed for Monte Carlo algorithms to reduce the initial probabilistic error by n^{-2} is of order $r = n^4$. The complexity of the Monte Carlo methods, for the investigated problems in this chapter, is $\mathcal{O}(n^3 \times r)$. To achieve the probabilistic error order $\mathbb{O}(n^{-2})$, the complexity of Monte Carlo methods has to be $\mathcal{O}(n^7)$. If we want to keep the error order at the level of $\mathcal{O}(n^{-2})$ and $\mathbb{O}(n^{-2})$ for Laplace's and Monte Carlo methods, respectively, the Monte Carlo methods are slightly more expensive than Laplace's method.

2.7 Simulation Results

In this section, we provide simulation results to evaluate the performances of different estimation methods. To this end, a realization of a Gaussian process that will serve as ground truth is shown in Fig. 2.2. The Gaussian process is generated for $\sigma_f = \sqrt{2}$ and $\sigma_0 = \sqrt{2}$ in (2.1). The measurement noise and the sampling position uncertainty variance are $\sigma_\epsilon = 0.01$ and $\Sigma_v = \sqrt{0.1} \times I$, respectively. $N = 20$ and $n = 40$ imply that 20 robot takes measurements twice at each sampling position. In Figs. 2.2-2.7, the predicted fields and predicted error variance fields are shown with color bars.

The results from Gaussian process regression using the noiseless positions $\mathbf{x}$ and the noisy measurement $\mathbf{y}$ are shown in Fig. 2.3.

To compare with typical quick-and-dirty solutions (QDS) to deal with noisy locations $\bar{\mathbf{x}}$ in practice, we define two solutions: QDS 1 and 2 approaches. QDS 1 is to applying Gaussian process regression given by (2.3) and (2.4) by simply ignoring noises in the locations and taking $\bar{\mathbf{x}}$ as the true positions, i.e.,

$$\mu^\star(\bar{\mathbf{x}}) = \mathbf{k}^T(\bar{\mathbf{x}})(\mathbf{K}(\bar{\mathbf{x}}) + {\sigma_\epsilon}^2 I)^{-1}\mathbf{y}. \tag{2.32}$$

When the measurements are taken repeatedly as suggested in Section 2.3, QDS 1 could be improved. In this regard, QDS 2 is to use the conditional expectation of sampling positions $\mathbf{x}$ given $\bar{\mathbf{x}}$ as in (2.19) and the least squares solution of $\mathbf{z}$ for given $\mathbf{y}$, which shall be plugged into Gaussian process regression, i.e.,

$$\mu^\star(H\bar{\mathbf{x}}) = \mathbf{k}^T(H\bar{\mathbf{x}})(\mathbf{K}(H\bar{\mathbf{x}}) + {\sigma_\epsilon}^2 I)^{-1}[(\bar{L}^T\bar{L})^{-1}\bar{L}^T\bar{\mathbf{z}}], \tag{2.33}$$

where H is from (2.19) and $\bar{L}$ is from (2.18). QDS 2 might be an improved version of QDS 1 when there are many repeated measurements for a set of fixed sampling positions. Figs. 2.4 and 2.5 show the results of applying QDS 1 and QDS 2 on $\bar{\mathbf{x}}$ and $\mathbf{y}$, respectively.

Table 2.2: Simulation RMS error

Estimator & Method	Figure	RMS error
Gaussian process regression in (2.3)	Fig. 2.3	0.1281
QDS 1 in (2.32)	Fig. 2.4	0.9126
QDS 2 in (2.33)	Fig. 2.5	0.7374
Monte Carlo method in (2.22)	Fig. 2.6	0.3403
Laplace's method in (2.27)	Fig. 2.7	0.3320
Simple Laplace method in (2.30)	Fig. 2.8	0.3503

This averaging helps the QDS approach to generate smoother predictions, and it shows improvement with respect to QDS 1.

The results of the Monte Carlo method with $r = 1000$ samples are shown in Fig. 2.6. The results of Laplace's method are shown in Fig. 2.7, and they look very similar to those of the Monte Carlo methods in Fig. 2.6. Figs. 2.4-2.7 clearly shows that both the Monte Carlo and Laplace's method outperform QDS 1 and 2 with respect to the true field in Fig. 2.3.

To numerically quantify the performance of each approach, we have computed the root mean square (RMS) error between the predicted and true fields for all methods, which are summarized in Table 2.2. This RMS error analysis could be done since we know the true realization of the Gaussian process exactly in this simulation study. As expected, Gaussian process regression with the true locations perform best. The proposed approaches, i.e., Monte Carlo and Laplace's method outperform QDS 1 and 2 in terms of RMS errors as well.

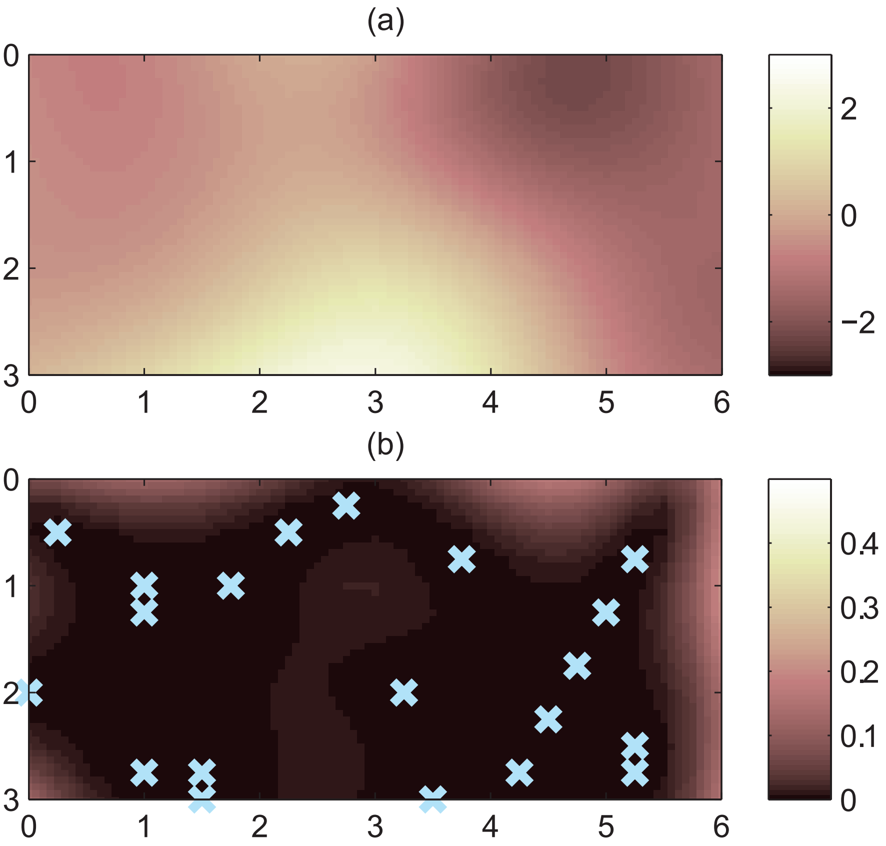


Figure 2.3: Gaussian process regression using the true positions $\mathbf{x}$. a) The predictive mean estimation. b) The predictive variance estimation, and the true sampling positions $\mathbf{x}$ in aquamarine crosses.

2.8 Experimental Results

2.8.1 Experiment with simulated noisy sampling positions

In this section, we apply proposed prediction algorithms to a real experimental data set to model the concentration of 7α, 12α, 24-trihydroxy-5α-cholan-3-one 24-sulfate (3kPZS), a synthesized sea lamprey (*Petromyzon marinus*) mating pheromone, in the Ocqueoc River, MI, USA which the authors of [49] provided. The sea lamprey is an ecologically damaging vertebrate invasive fish invader of the Laurentian Great Lakes [94] and a sea lamprey management program has been established [11]. A recent study by Johnson et al. [49] showed

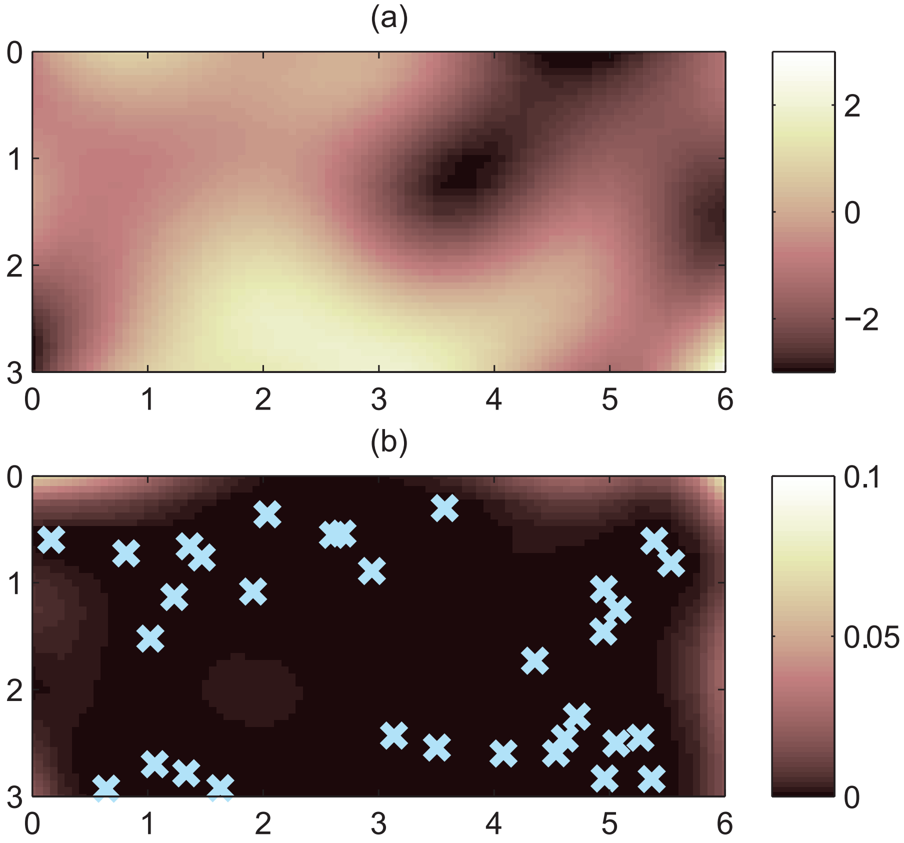


Figure 2.4: The results of the QDS 1 approximations using $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

that synthesized 3kPZS, a synthesized component of the male mating pheromone, when released into a stream to reach concentrations of $10^{-14} - 10^{-10}$ M (molar) or mol/L, triggers robust upstream movement in ovulated females drawing $\approx 50\%$ into baited traps. The ability to predict 3kPZS concentration at any location and time with a predicted uncertainty level would allow for fine-scale evaluations of hypothesized sea lamprey chemo-orientation mechanisms such as odor-conditioned rheotaxis [49]. Here 3kPZS was added to the stream to produce pulsing excitation to sea lampreys by applying 3kPZS at two minutes intervals [48]. To describe 3kPZS concentration throughout the experimental stream, rhodamine dye (Turner Designs, Rhodamine WT, Sunnyale, CA, USA) was applied at the pheromone release point (or source location) to reach a final in-stream concentration of 1.0 mug/L (mea-

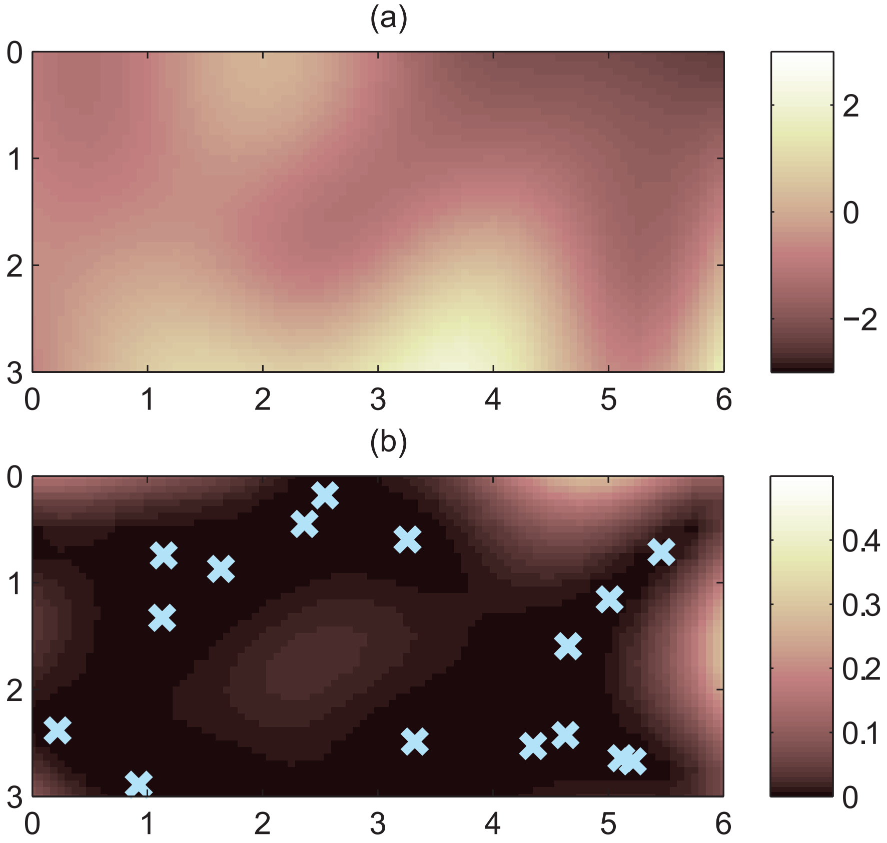


Figure 2.5: The results of the QDS 2 approximations using $\mathbb{E}(\mathbf{x}|\bar{\mathbf{x}})$. a) The predictive mean estimation. b) The predictive variance estimation, and $\mathbb{E}(\mathbf{x}|\bar{\mathbf{x}})$ shown with aquamarine crosses.

sure the concentration of the 3kPZS). The same pulsing strategy is used when 3kPZS was applied. The dye and 2kPZS pumping systems are shown in Fig. 2.9-(a). An example of the highly visible dye plume near the source location is shown in Fig. 2.9-(c).

To quantify dye concentrations in the stream, water samples were collected along transects occurring every 5 m from the sea lamprey release cage (0 m) to the application location (73 m upstream of release cage) as shown in Fig. 2.9-(b). Three sampling positions were fixed along each transect at 1/4, 1/2 and 3/4 of the channel width from the left bank. Water was collected from the three sampling sites along a transect simultaneously (three samplers) every 10 sec. for 2 minutes time interval. Further, a series of samples over 2 min were collected

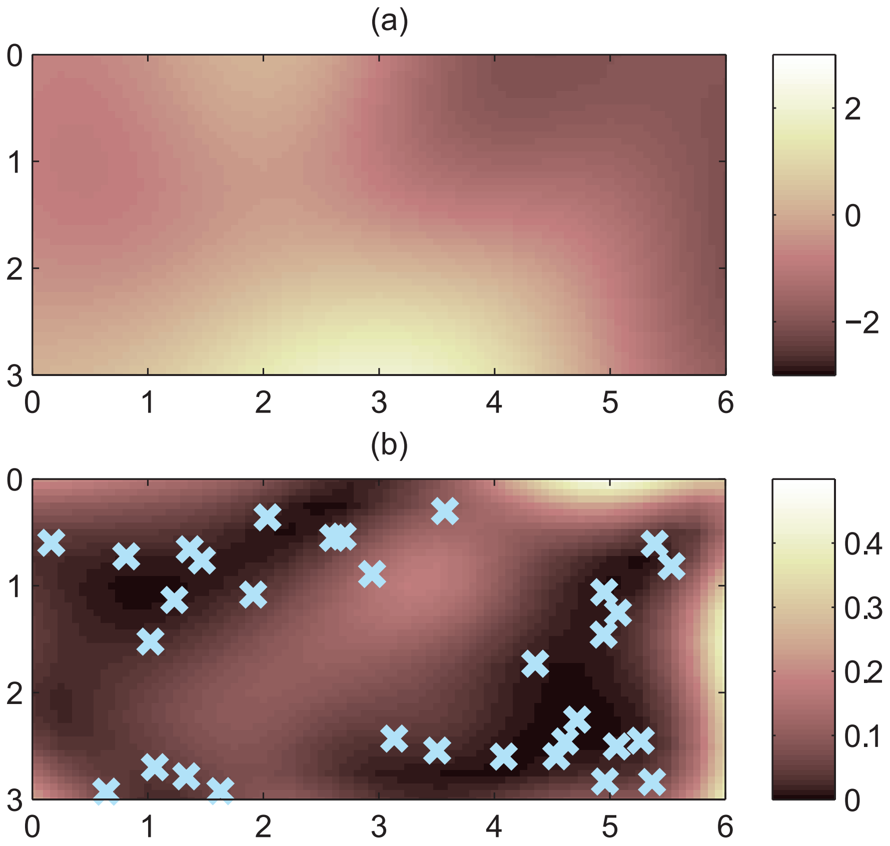


Figure 2.6: The results of the Monte Carlo approximations with 1000 samples using $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses..

exactly 1 meter downstream of the dye application location. Water samples were collected in 5 ml glass vials and subsequently the fluorescence intensity of each sample measured at 556 nm was determined in a luminescence spectrometer (Perkin Elmer LSS55, Downers Grove, IL, USA) and rhodamine concentration was estimated using a standard curve ($R_2 = 0.9998$).

The objective here is to predict the spatio-temporal field of the dye concentration. In fact, the sampling positions are exactly known from the experiment. Therefore, we will intentionally inject some noises in the sampling positions to evaluate the proposed prediction algorithms in this chapter. Before applying the proposed algorithms, the hyperparameters such as σ_f and σ_0 were identified from the experimental data by a maximum a posteriori

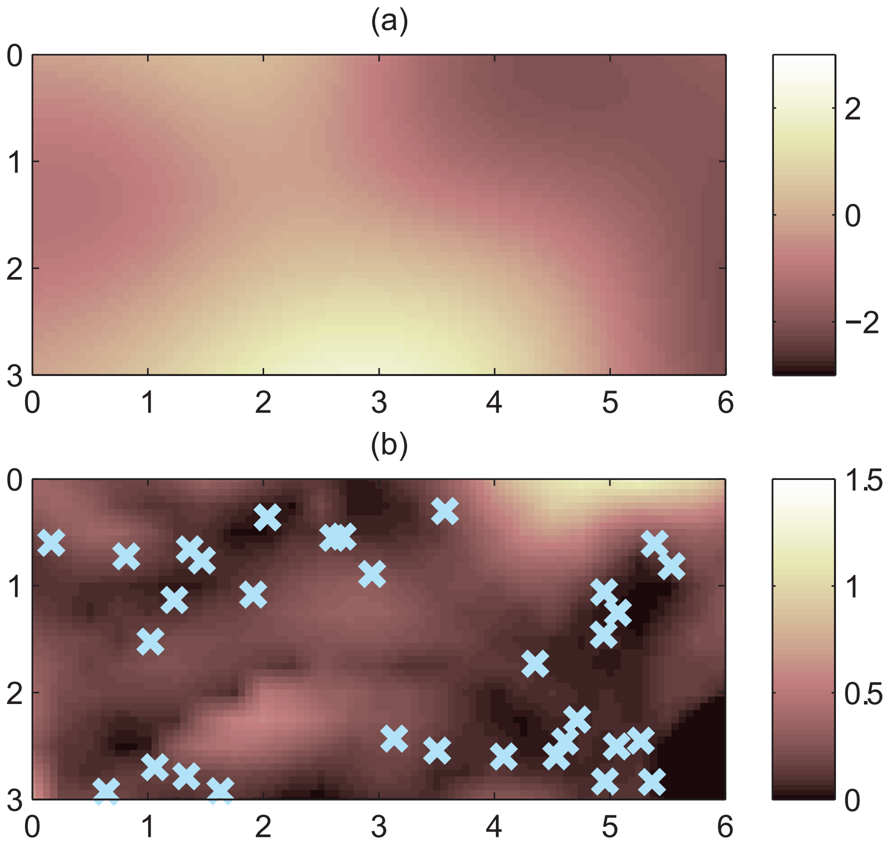


Figure 2.7: The results of the Laplace approximations using $\bar{\mathbf{x}}$. a) Predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

probability (MAP) estimator as described in [113]. On the other hand, the value for σ_ϵ was set to $\sigma_\epsilon = 5 \times 10^{-3}$ mug/L according to the noise level from the data sheet of the sensor. We consider the anisotropic covariance function [113] to deal with the case that the correlations along different directions in $x = [x_1, x_2, x_3]^T \in \mathbb{R}^3$ are different. Next, we performed a change of variables such that in a transformed space and time. The covariance function, given (2.1), could be used for the proposed approaches.

For the case of this experiment, the spatial correlation along the river flow direction is different from the correlation perpendicular to the river flow direction. We consider the

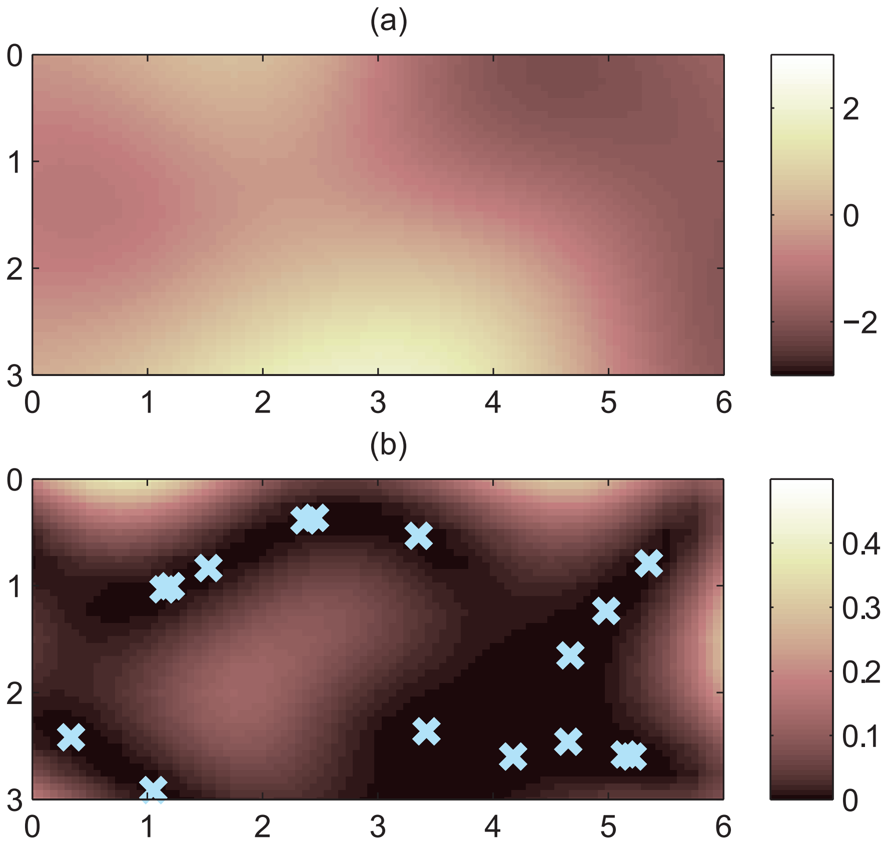


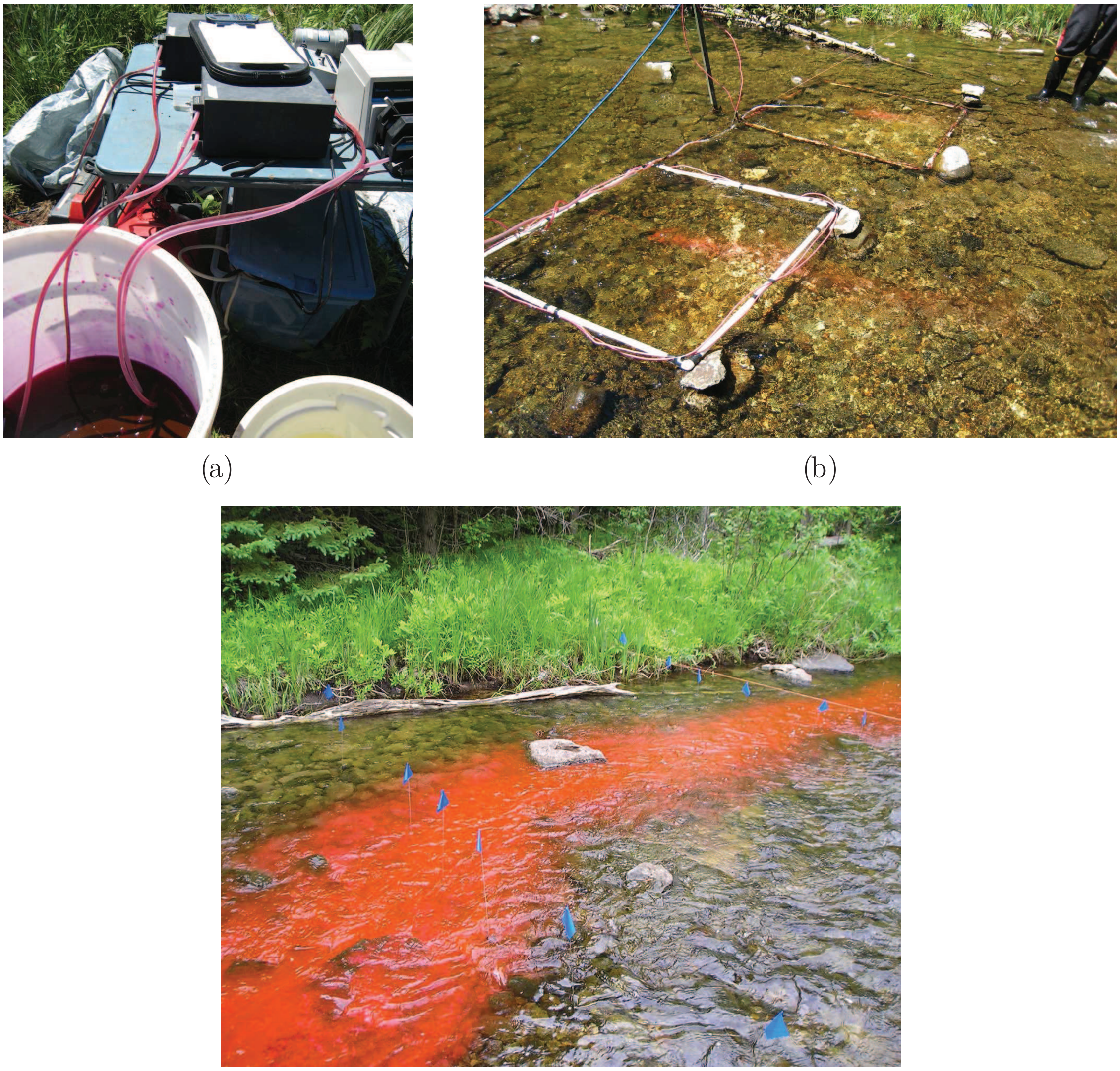
Figure 2.8: The results of the simple Laplace approximations using $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\hat{\mathbf{x}}$ shown with aquamarine crosses. $\hat{\mathbf{x}}$ is the estimation of the true sampling positions, computed as a by-product of both fully exponential Laplace and simple Laplace approximations.

following covariance function in the units for the experimental data.

$$\mathcal{K}(x, x') = {\sigma_f}^2 \exp\left(-\sum_{j=1}^{2} \frac{(x_j - x'_j)^2}{2\sigma_j^2}\right). \tag{2.34}$$

Using this covariance function in (2.34) and computing likelihood function with true value of $\mathbf{x}$, the hyperparameter vector $\theta := [\sigma_f, \sigma_1, \sigma_2]$ can be computed by the MAP estimator as follow:

$$\theta_{MAP} = \arg\max_{\theta \in \Theta} \mathbb{P}(\mathbf{y}|\theta)\, \mathbb{P}(\theta). \tag{2.35}$$



(a)

(b)

(c)

Figure 2.9: (a) The dye pumping system. (b) An example of normal dye concentration applied to the stream for the pheromone distribution estimation. (c) An example of the highly visible dye plume near the source location.

Table 2.3: Experiment parameters

Description	Parameter	Value
Number of sensors	N	15
Number of measurements	n	15
The variability at a fixed point	σ_f	$\sqrt{0.3}$
Bandwidth	σ_1 (vertical)	1.6
	σ_2 (horizontal)	10.7
Noise covariance matrix	Σ_v	$\begin{bmatrix} 4 & 0 \\ 0 & 0.089 \end{bmatrix}$
Measurement noise level	σ_ϵ	0.005

Using the optimization in (2.35), we obtained $\sigma_f = \sqrt{0.3}$ mug/L, $\sigma_1 = 1.6$ m and $\sigma_2 = 10.7$ m. x_1 and x_2 are the coordinates of the vertical and horizontal (flow direction) axes of Fig. 2.10. As we expect, we have $\sigma_1 < \sigma_2$, i.e., the field is more correlated along the river flow direction as compared to the perpendicular to the river flow direction. Next, after finding MAP estimates of hyperparameters σ_1 and σ_2, we change the scale of each component of x, such that we have the same values for hyperparameters in the transformed coordinates. The new coordinates are given by $\{x_i^{new} = \frac{\sigma_0}{\sigma_i} x_i\}$ and we recover

$$\mathcal{K}(x, x') = {\sigma_f}^2 \exp\left(-\sum_{i=1}^{b} \frac{\left((\frac{\sigma_0}{\sigma_i})x_i - (\frac{\sigma_0}{\sigma_i})x'_i\right)^2}{2\sigma_0^2}\right),$$

as in (2.1). Note that scaling x_i with $\frac{\sigma_0}{\sigma_i}$, also transforms the covariance matrix Σ_v in a normalized space. Table 2.3 shows parameter values which are estimated and used for the experimental data.

Fig. 2.10 shows the predicted field by applying Gaussian process regression on the true positions $\mathbf{x}$. To illustrate the advantage of proposed methods over the QDS approach in dealing with the noisy localizations, we show the results of applying QDS 1, Monte Carlo, Laplace, and simple Laplace approximations in Figs. 2.11, 2.12, 2.13, and 2.14, respectively.

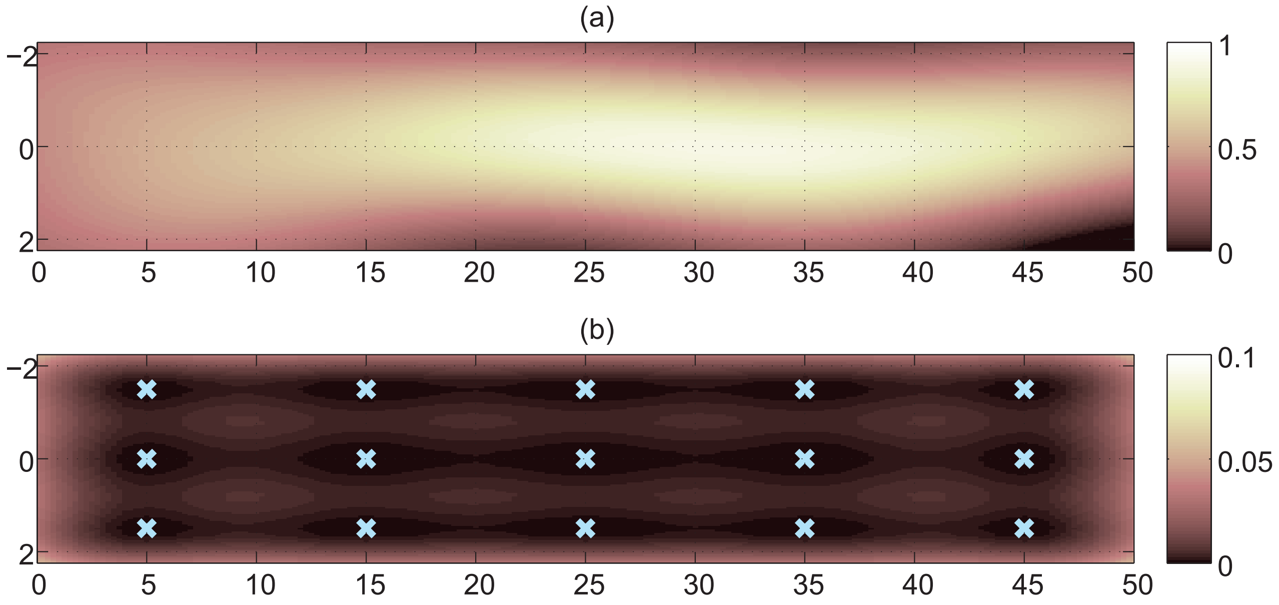


Figure 2.10: The results of Gaussian process regression using true $\mathbf{x}$. a) The posterior mean estimation. b) The posterior variance estimation, and $\mathbf{x}$ shown with aquamarine crosses.

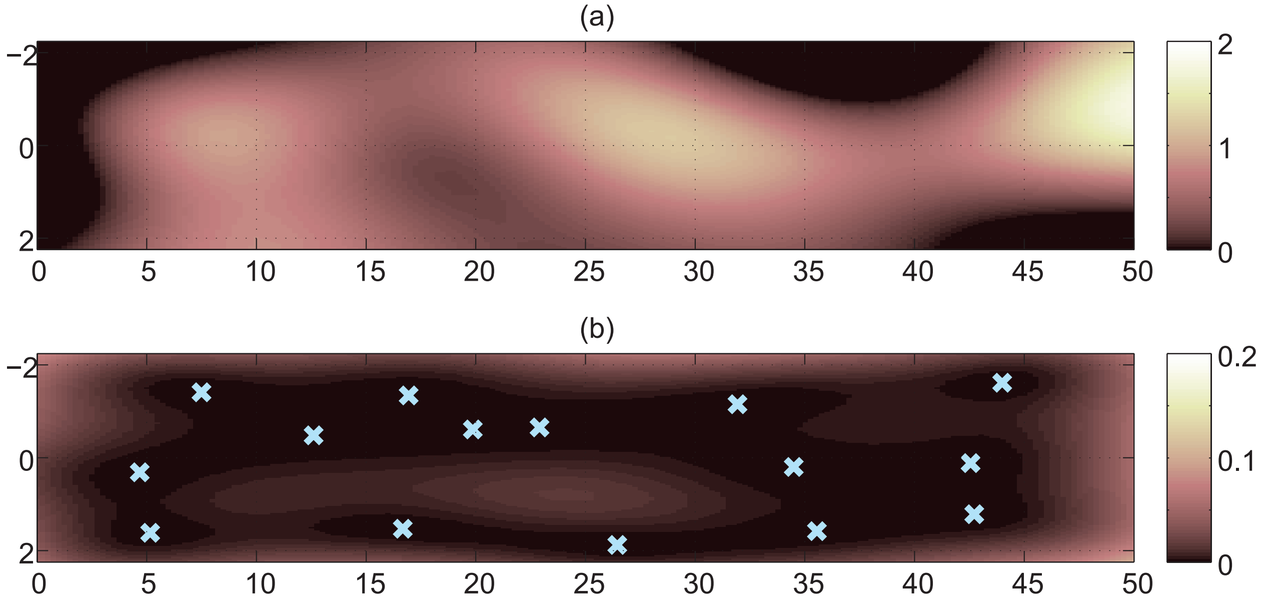


Figure 2.11: The results of the QDS 1 approach using noisy positions $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

Since there is only one measurement per position, QDS 2 is same as QDS 1. As can be seen in these figures, the peaks of the predicted fields by Monte Carlo and Laplace's method

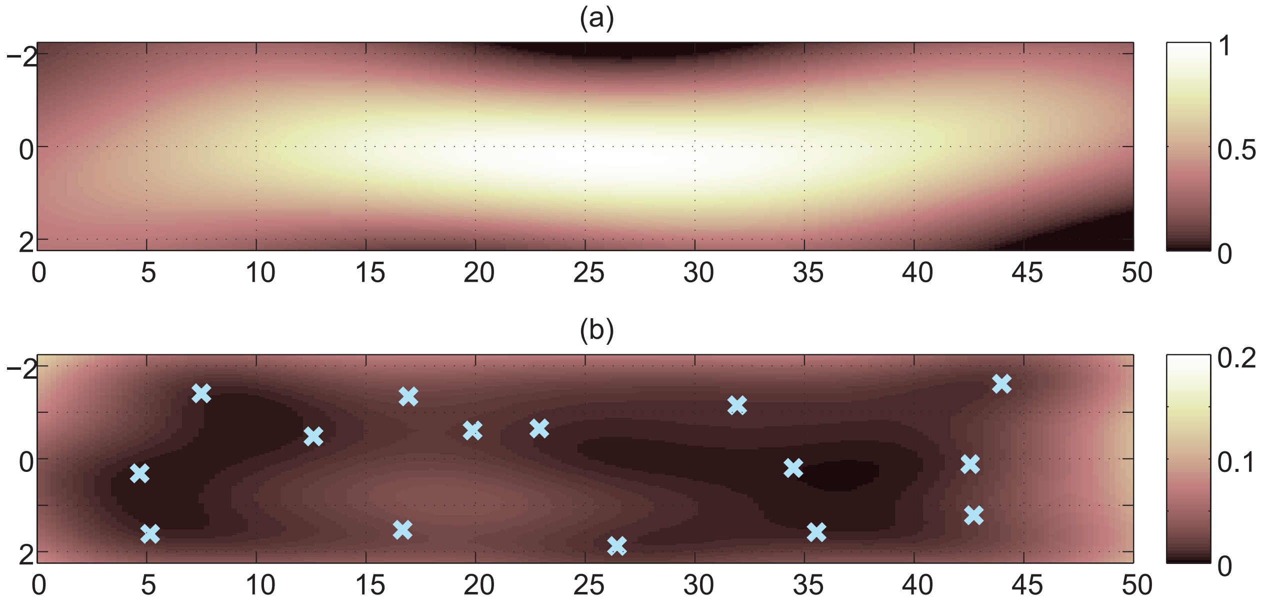


Figure 2.12: The results of the Monte Carlo method with 1000 samples using noisy positions $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

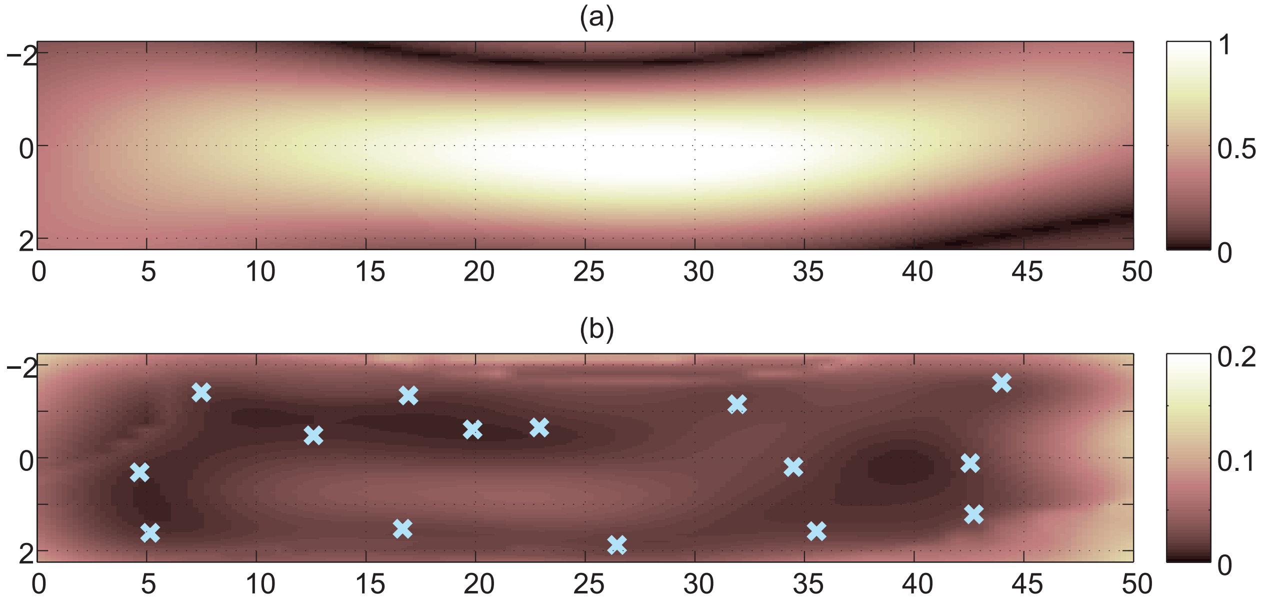


Figure 2.13: The results of the Laplace method using noise augmented positions $\bar{\mathbf{x}}$ shown with aquamarine crosses. a) The predictive mean estimation. b) The predictive variance estimation and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

with noisy $\bar{\mathbf{x}}$ are closer to the peak predicted by the true $\mathbf{x}$ than that of the QDS approach. Indeed, QDS 1 has failed in Fig. 2.11 to produce a good estimation of the true field with

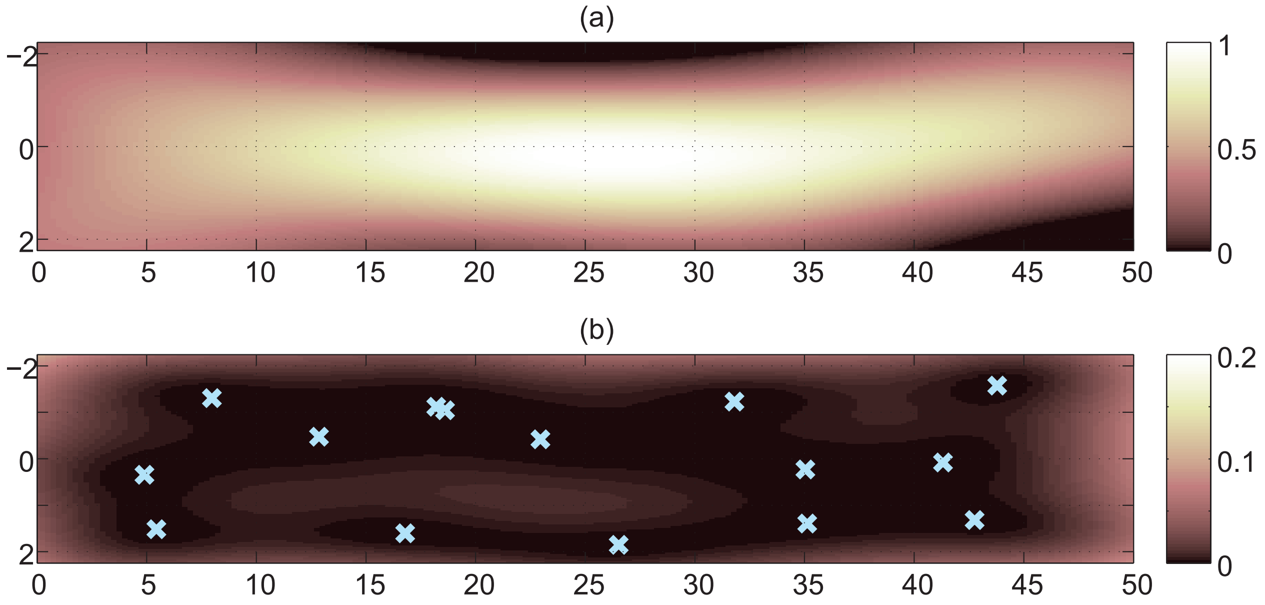


Figure 2.14: The results of the simple Laplace method using noise augmented positions $\bar{\mathbf{x}}$ shown with aquamarine crosses. a) The predictive mean estimation. b) The predictive variance estimation and $\hat{\mathbf{x}}$ shown with aquamarine crosses.

Table 2.4: Experimental RMS error w.r.t. Gaussian estimation

Estimator & Method	Figure	RMS error
QDS 1 in (2.32)	Fig. 2.11	0.3955
Monte Carlo method in (2.22)	Fig. 2.12	0.1176
Laplace's method in (2.27)	Fig. 2.13	0.1305
Simple Laplace method in (2.30)	Fig. 2.14	0.1252

neglecting the uncertainty in the positions, while Monte Carlo, Laplace and simple Laplace methods, proposed in this chapter, produce good estimations in Figs. 2.12, 2.13, and 2.14, respectively.

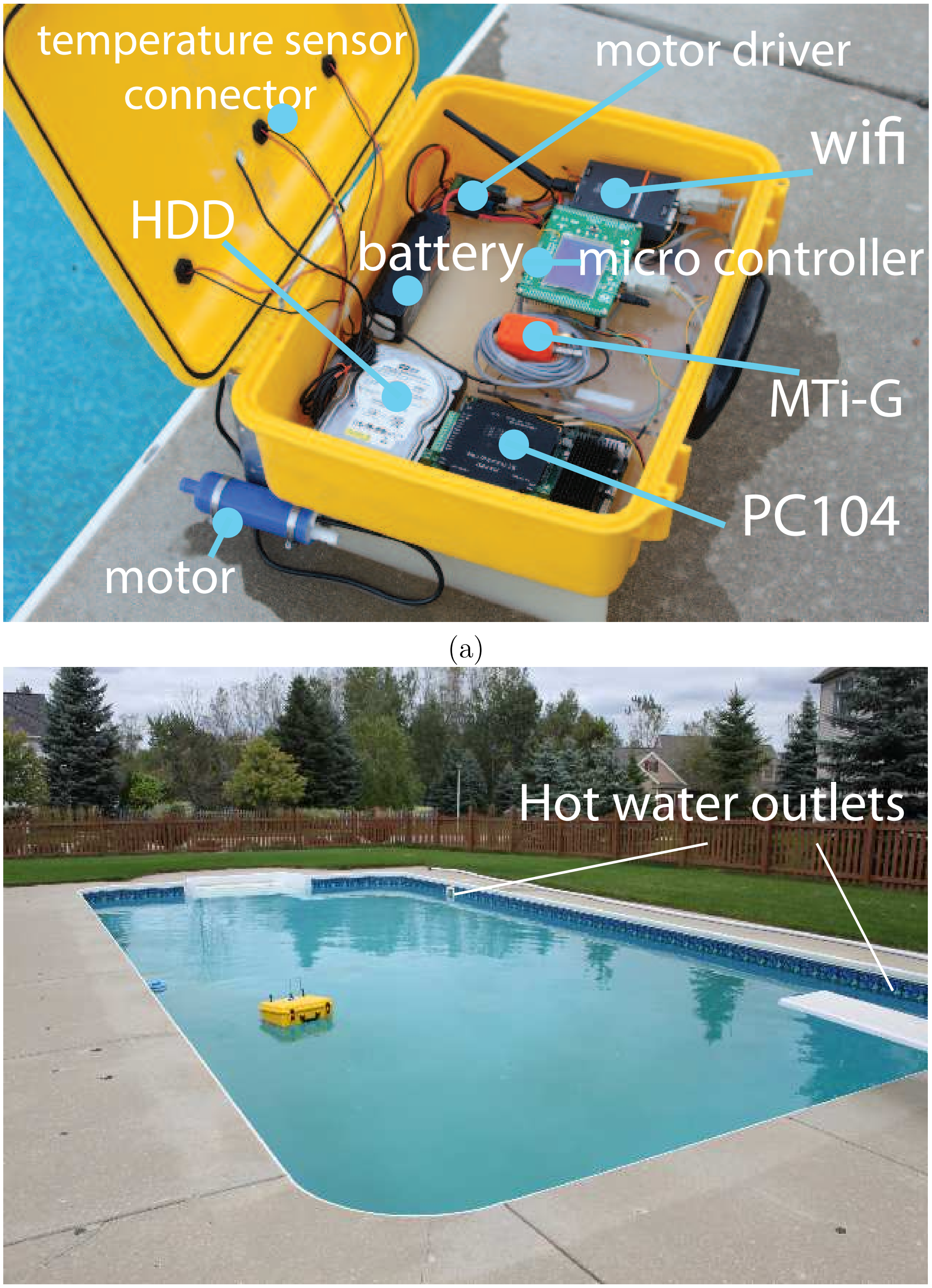
Gaussian regression using true $\mathbf{x}$ is our best estimation of the true field (Fig. 2.10). Table 2.4 shows the RMS errors of the different estimation approaches using the noisy $\mathbf{x}$ (i.e. $\bar{\mathbf{x}}$) with respect to Gaussian regression using the true $\mathbf{x}$. As can be seen in Table 2.4, the proposed approaches outperform QDS 1 in terms of the RMS error.

2.8.2 Experiment with noisy sampling positions

In Section 2.8.1 we used the true and simulated noisy sampling positions to compare the performance of the proposed approaches. In this section, we present another set of experimental results under real localization errors. The experimental results were obtained using a single aquatic surface robot in an outdoor swimming pool as shown in Fig. 2.15. The robot is capable of monitoring the water temperature in an autonomous manner while could be remotely supervised by a central station as well. To measure the location of the robot, we used a Xsense MTi-G sensory unit [1] (as shown in the center of Fig. 2.15-(a)) which consists of an accelerometer, a gyroscope, a magnetometer, and a Global Positioning System (GPS) unit. A Kalman filter is implemented inside of this sensor by the company, which produces the localization of the robot with accuracy of one meter. More information about this experiment can be found in [39]. In this experiment, we have selected and identified the system parameters based on a priori knowledge about the process and sensor noise characteristics. Therefore, model hyperparameters such as $\sigma_0 = 2$, $\Sigma_v = I$, $\sigma_\epsilon = 0.005$, and ${\sigma_f}^2 = 0.3$ are known. The number of sampling positions and sampled measurements are $N = n = 10$.

To validate our approaches, we have controlled the temperature field by turning the hot water pump on and off. The hot water outlet locations have been shown in Fig. 2.15-(b). We turned on the hot water pump for a while. After that, the hot water pump was turned off, and after 6 minutes, the robot collected 10 measurements with 10 sec. time intervals.

The estimated temperature and its error variance fields, by applying QDS 1, Monte Carlo, fully exponential Laplace, and simple Laplace methods are shown in Figs. 2.16, 2.17, 2.18, and 2.19, respectively. For each method, the estimated temperature and its error variance fields are shown in subfigures of (a) and (b), respectively.



(a)

(b)

Figure 2.15: a) The developed robotic sensor. b) The experimental environment- a 12 × 6 meters outdoor swimming pool.

As can be seen in Figs. 2.17, 2.18, and 2.19, results from Monte Carlo, fully exponential Laplace, and simple Laplace methods are well matched.

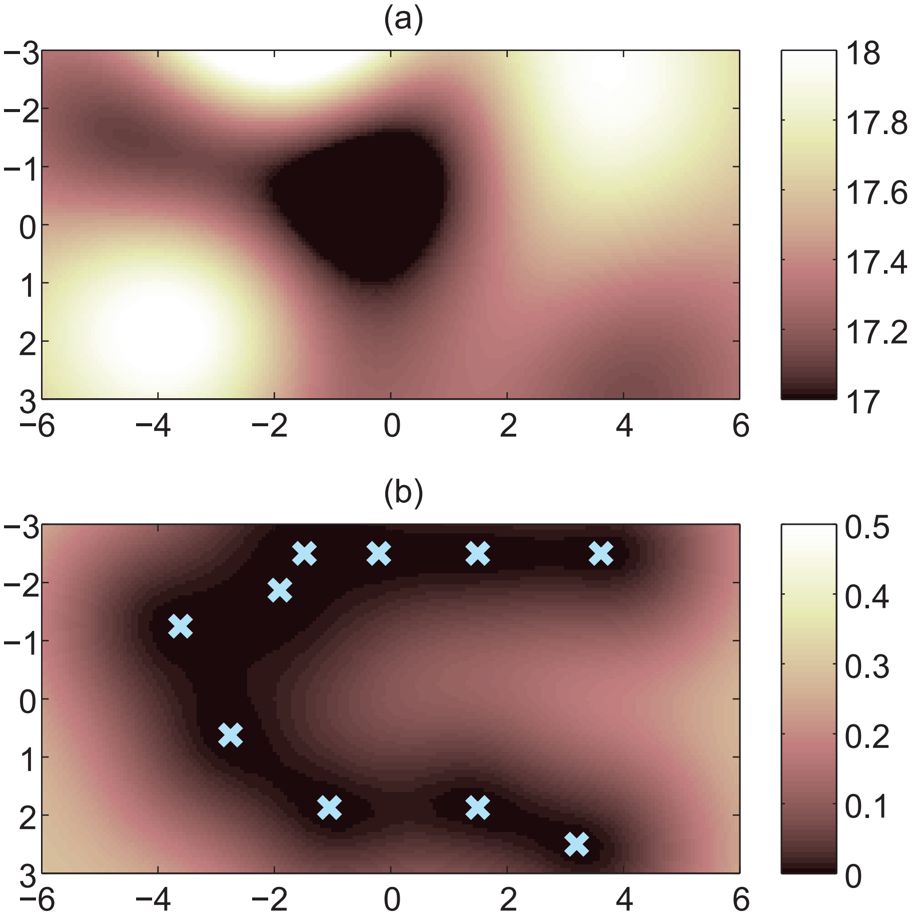


Figure 2.16: The results of the QDS 1 approach using noisy positions $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

2.9 Conclusion

We have formulated Gaussian process regression with observations under the localization uncertainty due to (possibly mobile) sensor networks with limited resources. Effects of the measurements noise, localization uncertainty, and prior distributions have been all correctly incorporated in the posterior predictive statistics in a Bayesian approach. We have reviewed the Monte Carlo sampling and Laplace's method, which have been applied to compute the analytically intractable posterior predictive statistics of the Gaussian processes with localization uncertainty. The approximation error and complexity of all the proposed approaches

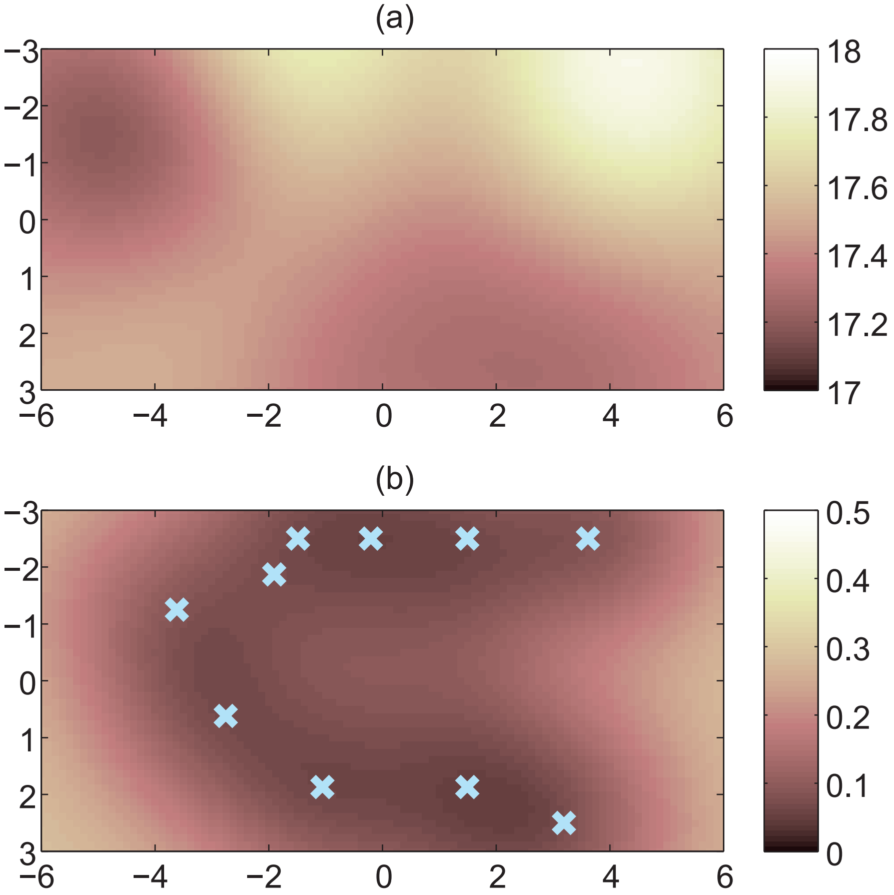


Figure 2.17: The results of the Monte Carlo method with 1000 samples using noisy positions $\bar{\mathbf{x}}$. a) The predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

have been analyzed. In particular, we have provided tradeoffs between the error and complexity of Laplace approximations and their different degrees such that one can choose a tradeoff taking into account the performance requirements and computation complexity due to the resource-constrained sensor network. Simulation study demonstrated that the proposed approaches perform much better than approaches without considering the localization uncertainty properly. Finally, we applied the proposed approaches on the experimentally collected real data to provide proof of concept tests and evaluation of the proposed schemes in practice. From both simulation and experimental results, the proposed methods outper-

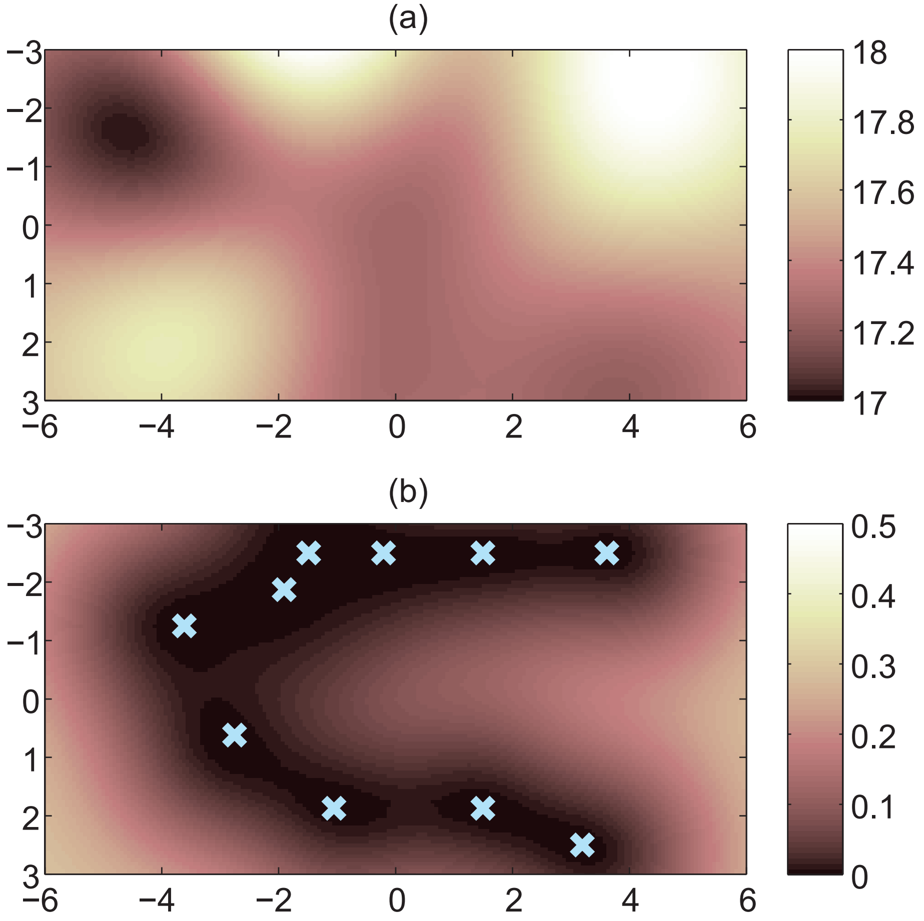


Figure 2.18: The results of the Laplace approximations using $\bar{\mathbf{x}}$. a) Predictive mean estimation. b) The predictive variance estimation, and $\bar{\mathbf{x}}$ shown with aquamarine crosses.

formed the quick-and-dirty solutions often used in practice.

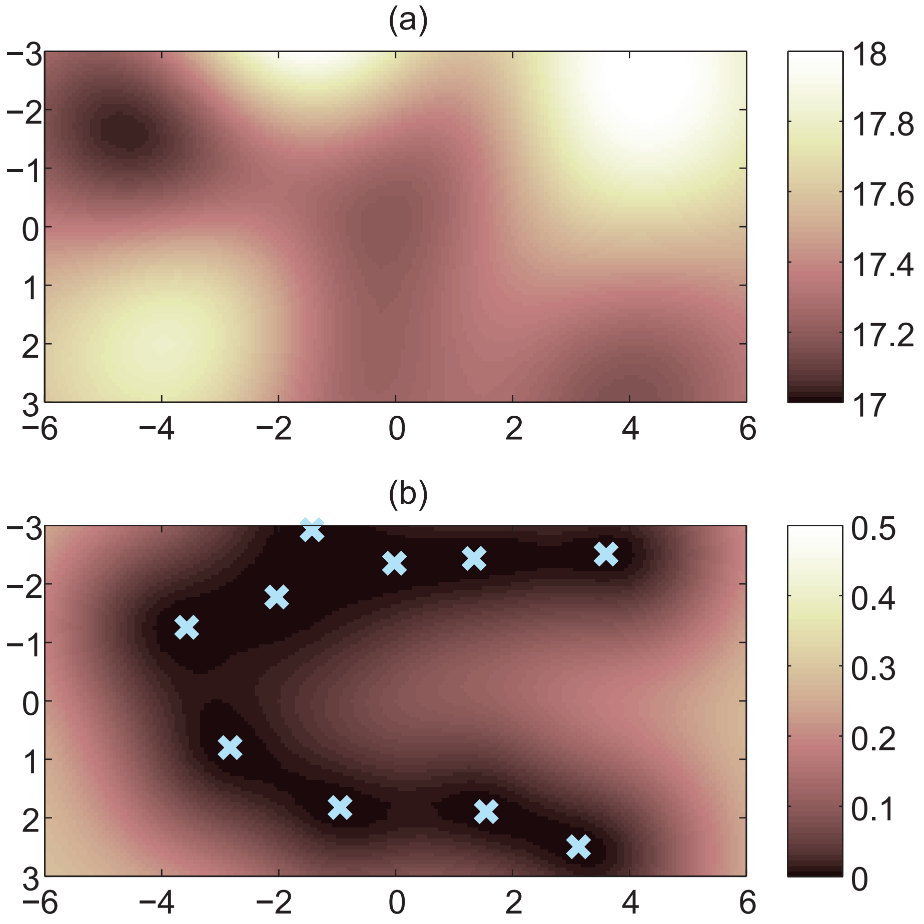


Figure 2.19: The results of the simple Laplace method using noise augmented positions $\bar{\mathbf{x}}$ shown with aquamarine crosses. a) The predictive mean estimation. b) The predictive variance estimation and $\hat{\mathbf{x}}$ shown with aquamarine crosses.

Chapter 3

Position estimation using Gaussian process regression

Abstract: This chapter presents a position estimation method for a robot using an omnidirectional camera. We present an approach to build a map from optimally selected visual features using Gaussian Process (GP) regression. Optimal feature selection for regression is challenging in general. In this chapter, we apply a cross-validation technique to select features in order to improve the quality of the position estimation. The position of the robot is estimated by a maximum likelihood estimation (MLE). The collection of selected features over a surveillance region is modeled by a multivariate GP with unknown hyperparameters. The hyperparameters are identified through the learning process by an MLE, which are used in prediction in an empirical Bayes approach. The excellent results of the proposed algorithm are illustrated by the experimental study under two different real-world scenarios with and without uncertainty in sampling positions.

3.1 Introduction

Localization of a mobile robot relative to its environment using vision information has received extensive attention over past few decades from the robotic and computer vision communities [5, 20]. Vision-based robot positioning involves two steps. The first step involves learning some properties of vision data (features) with respect to the spatial position where observation is made, so-called mapping. The second step is to find the best match for the new spatial position corresponding to the newly observed features, so-called matching. The mapping from these visual features to the domain of the associated spatial position is highly nonlinear and sensitive to the type of selected features. In most cases, it is very difficult to derive the map analytically. The features shall vary as much as possible over the spatial domain while varying as small as possible for a given position over the disturbance. For example, they should be insensitive to changes in illumination and partial obstruction.

Motivated by the aforementioned situations, we consider the problem of selecting features from the original feature set in order to improve the localization performance of a robot. The central assumption when using a feature selection technique is that the original feature set contains many redundant or irrelevant features. To facilitate further discussion, let us consider a configuration where the input vector X is the robot position and the output feature vector Y is the collection of extracted features from the vision data. We first build a feature map F at a robot location X such that $F(X) = Y$. In order to reduce position estimation error, the ideal subset is defined as follows.

$$Y_{opt} = \arg\min_{\hat{Y}} \|X - F^{-1}(\hat{Y})\|^2,$$

where $\hat{Y}$ is a vector that consists of the selected entries of the original vector Y. However with a high cardinality of the original feature set, the optimal solution relies on the combinatorial optimization which is not feasible.

On the other hand, using mutual information criteria, F and F^{-1} could be chosen as follows:

$$F(X) = \arg\max_{Y} I(X,Y), \quad F^{-1}(Y) = \arg\max_{X} I(X,Y),$$

where $I(X,Y) = \int\int \mathbb{P}(X,Y)\log\left(\frac{\mathbb{P}(X,Y)}{\mathbb{P}(X)\,\mathbb{P}(Y)}\right)$ is the mutual information of X and Y. Note that, in the case where $\mathbb{P}(X)$ and $\mathbb{P}(Y)$ are constant then $F(X)$ is equivalent to the maximized log-likelihood function. Guo et al. [33] show by using mutual information, one can achieve a recognition rate higher than 90% while just using 0.61% of feature space for a classification problem. The only issue with mutual information is with its computational complexity and to avoid that, one has to avoid the higher dependencies among variables [81].

In order to make a fast and precise estimation, most of the existing localization algorithms extract a small set of important features from the robotic sensor measurements. The features used in different approaches for robotic localization range from *C.1*: artificial markers such as color tags [45] and barcodes (that need to be installed) [92], *C.2*: geometric features such as straight wall segments and corners [46], and to *C.3*: natural features such as light and color histograms [4]. Most of the landmark-based localization algorithms are classified in *C.1* and *C.2*. It is shown in [87] that autonomous navigation is possible for outdoor environments with the use of a single camera and natural landmarks. In a similar attempt, [22] addressed the challenging problem of indoor place recognition from wearable video recording devices.

The localization methods which rely on artificial markers (or static landmarks) have three disadvantages: lack of flexibility, lack of optimality, and lack of autonomy. A method is described in [99] that enables robots to learn landmarks for localization. Artificial neural networks are used for extracting landmarks. However, the localization methods which rely on dynamic landmarks [99] have disadvantages such as lack of stability. Furthermore, there are reasons to avoid the geometric model as well, even when a geometric model does exist.

Such cases may include: 1) the difficulty of reliably extracting sparse, stable features using geometrical models, 2) the ability to use all sensory data directly rather than a relatively small amount of abstracted discrete information obtained from feature extraction algorithms, and 3) high computational and storage costs of dealing with dense geometric features.

In contrast to the localization problem with artificial markers or popular geometrical models, there is a growing number of practical scenarios in which global statistical information is used instead. Some works illustrate localization using various spatially distributed (continuous) signals such as distributed wireless Ethernet signal strength [27], or multi-dimensional magnetic fields [105]. In [110], a neural network is used to learn the implicit relationship between the pose displacements of a 6-DOF robot and the observed variations in global descriptors of the image, such as geometric moments and Fourier descriptors. In similar studies, gradient orientation histograms [55] and low dimensional representation of the vision data [103] are used to localize mobile robots. In [80], an algorithm is developed for navigating a mobile robot using a visual potential. The visual potential is computed from the image appearance sequence captured by a camera mounted on the robot. A method for recognizing scene categories by comparing the histograms of local features is presented in [59]. Without explicit object models, by using global cues as indirect evidence about the presence of an object, they consistently achieve an improvement over an orderless image representation [59].

The recent research efforts that are closely related to our problem are summarized as follows. The location for a set of image features from new observations is inferred by comparing new features with the calculated map [7, 66, 67]. In [109], a neural network is used to learn the mapping between image features and robot movements. Similarly, there exists effort on automatically finding the transformation that maximizes the mutual information between two random variables [107]. Using Gaussian process (GP) regression, the authors of [7, 91] present effective approaches to build a map from a sparse set of noisy observations taken from known locations using an omnidirectional camera. While the selection of visual features for such applications determines the ultimate performance of the algorithms, such a topic has not been investigated to date. Therefore, building on Brook's approach [7] our work expands it more on the feature extraction and selection in order to improve the quality of localization. A Bayesian point of view is taken to make the map using a GP framework.

The contributions of the chapter are as follows. This chapter provides a position estimation method for a robot using an omnidirectional camera. We present an approach to build a map from optimally selected visual features using GP regression. First, we describe how we can extract some robust properties from vision data captured by an omnidirectional camera (Section 3.2). In particular, we describe how the fast Fourier transform (FFT) is applied to the panoramic image to calculate a set of image properties. We then transform the high dimensional vision data to a set of uncorrelated feature candidates. A multivariate GP regression with unknown hyperparameters is formulated to connect the set of selected features to their corresponding sampling positions (Section 3.3). An empirical Bayes method using a point estimate is used to predict the feature map. Next, a feature reduction approach is developed using a cross-validation method such that an optimal subset of the features is selected to minimize the Root Mean Square Error (RMSE) (Section 3.4). The effectiveness of the proposed algorithms is illustrated by excellent experimental results under two different setups (Section 3.5). Additionally, some comparison results are presented under different images features and feature selection methods.

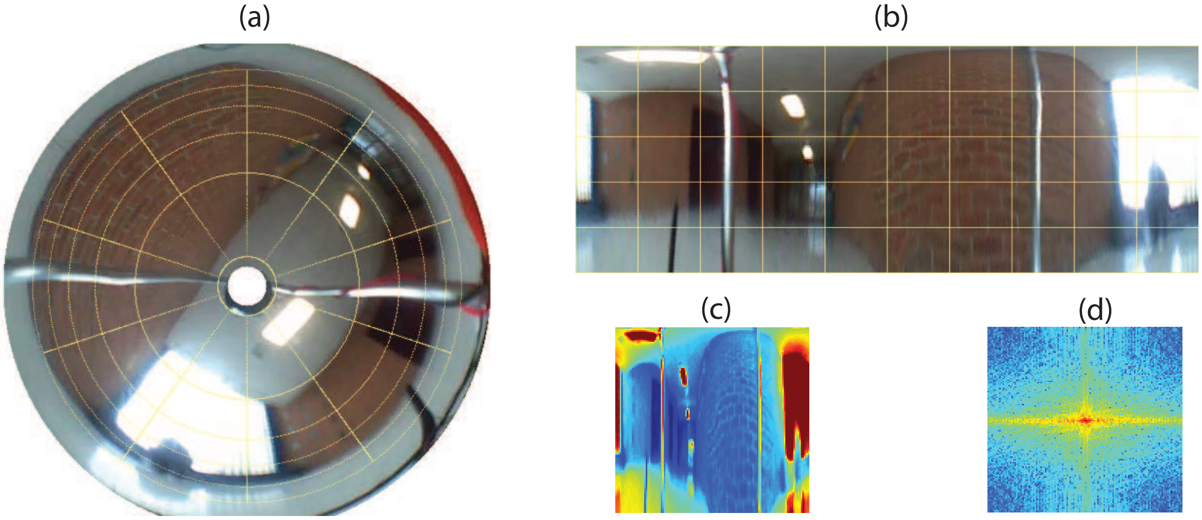


Figure 3.1: (a) and (b) show the wrapped omnidirectional image and the unwrapped panoramic image, respectively. (c) and (d) show the reduced size gray scale image and the two-dimensional FFT magnitude plot, respectively.

3.2 Image features

Conventional video cameras with projective lens have restricted fields of view. Adding mirrors of different shapes, 360° panoramic views can be achieved in a single image [26]. In this chapter, to make localization insensitive to heading angle, a spherical mirror is used to capture a 360° view from the environment of a robot.

Before an omnidirectional image is processed, it is first unwrapped. When it comes from the camera, the image is a nonlinear mapping of a 360° panoramic view onto a donut shape. Recovering the panoramic view from the wrapped view requires the reverse mapping of pixels from the wrapped view onto a panoramic view [19, 66]. Figs. 3.1-(a) and 3.1-(b) show the wrapped omnidirectional image and the unwrapped panoramic image, respectively.

The fast Fourier transform (FFT) is applied to the panoramic image to calculate a set of image properties y. For a square image of size $N \times N$, the two-dimensional FFT is given by

$$F^{[i]}(\rho, l) = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f^{[i]}(a, b) e^{-j2\pi(\rho \frac{a}{N} + l \frac{b}{N})},$$

where $f^{[i]}$ is the i-th two-dimensional realized image, and j is the imaginary unit. In order to use FFT, we convert panoramic color images to gray scale 128×128 pixel images, i.e., $[f(a, b)]$. Fig. 3.1-(c) and (d) show the reduced size gray scale image and its two-dimensional FFT magnitude plot, respectively. The Fourier transform is an important image processing tool which is used to decompose an image into its periodic components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is in the spatial domain. In image processing, often only the magnitude of the Fourier transformed image is utilized, as it contains most of the information of the geometric structure of the spatial domain image [76]. The Fourier transform is used in a wide range of applications such as image analysis, image filtering, image reconstruction, and image compression. In addition, localization methods are proposed in [66, 110] where every image is represented by a family of components in the Fourier domain.

We briefly highlight three properties obtained by applying the Fourier transform on the omnidirectional images. First, the magnitude of the Fourier components can be represented by a continuous function of the robot position. Next, the phase angles of the Fourier components are related to the heading of the robot. Finally, by representing an image with a small number of frequency components in the Fourier domain, data compression can be achieved. In [66], it was shown that the first 15 components of FFT carry enough information to correctly match a pair of images.

In an indoor environment most of the objects stand vertically. In this planer navigation case, it is reasonable to assume that features in the horizontal direction are more important than those in the vertical direction. For example, the borders between wall and doors and widows in the horizontal direction will be observed by this planar robot. As expected, from the spectrum Fig. 3.1-(c) it can be seen that almost all the information is contained near the center and the low parts of the frequencies. It is reasonable that components in the frequencies provide the best ground for the localization process. Therefore, the image properties are computed by:

$$y_\rho^{[i]} = \left|F^{[i]}(\rho, 0)\right| = \left|\sum_{b=0}^{127}\left(\sum_{a=0}^{127} f^{[i]}\left(a, b\right)\right) e^{\frac{-j2\pi\rho b}{128}}\right|, \tag{3.1}$$

where $|\cdot|$ denotes the absolute value of a complex number.

Equation (3.1) can be interpreted as the one-dimensional FFT of the image that is averaged over the vertical direction.

In addition to the FFT coefficients, the histogram for the image, and wavelet coefficients could be used as image properties [59]. An image histogram is a type of the histogram that acts as a graphical representation of the tonal distribution in a digital image. The number of different tonal bins in the histogram is specified by the number of properties. Then the histogram determines the number of pixels for each tonal value. A discrete wavelet transform (DWT) is a wavelet transform for which the wavelets are discretely sampled. The wavelet coefficients is also used widely by appearance-based place recognition methods [7, 103]. In Section 3.5, we will experimentally evaluate our approach with features obtained by using FFT, histogram, and discrete wavelet transforms.

3.3 Gaussian process (GP) model

We propose a multivariate GP as a model for the collection of image features. A GP defines a distribution over a space of functions and it is completely specified by its mean function and covariance function. We denote that $y_\rho^{[i]} := y_\rho\left(s^{[i]}\right) \in \mathbb{R}$ is the i-th realization of the ρ-th image property and $s^{[i]} \in \mathcal{S}$ is the associated position where the realization occurs. Here $\mathcal{S}$ denotes the surveillance region, which is a compact set. Then, the accumulative image properties y is a random vector defined by $y = \left(y_1^T, \cdots, y_m^T\right)^T \in \mathbb{R}^{nm}$, and $y_\rho = \left(y_\rho^{[1]}, \cdots, y_\rho^{[n]}\right) \in \mathbb{R}^n$ contains n realizations of the ρ-th image property.

We assume that the accumulative image properties can be modeled by a multivariate GP, i.e. $y \sim \mathcal{GP}(\Gamma, \Lambda)$, where $\Gamma : \mathcal{S}^n \to \mathbb{R}^{mn}$ and $\Lambda : \mathcal{S}^n \to \mathbb{R}^{mn \times mn}$ are the mean function and the covariance function, respectively. However, the size and multivariate nature of the data lead to computational challenges in implementing the framework.

For models with multivariate output, a common practice is to specify a separable covariance structure for the GP for efficient computation. For example, Higdon [36] calibrated a GP simulator with the high dimensional multivariate output, using principal components to reduce the dimensionality. Following such model reduction techniques, we transform the vector y to a vector z such that its elements $\left\{z_\rho | \rho \in \Omega_m\right\}$, where $\Omega_m = \{1, \cdots, m\}$ are i.i.d.

The statistics of y can be computed from the learning dataset.

$$\mu_y = \frac{1}{2}\sum_{i=1}^{n} y^{[i]}, \quad \Sigma_y = \frac{1}{n-1}\sum_{i=1}^{n} ||y^{[i]} - \mu_y||^2.$$

The singular value decomposition (SVD) of Σ_y is a factorization of the form $\Sigma_y = USU^T$, where U is a real unitary matrix and S is a rectangular diagonal matrix with nonnegative real numbers on the diagonal. In summary, the transformation will be performed by the following formula.

$$z^{[i]} = S^{-1/2}U^T(y^{[i]} - \mu_y). \tag{3.2}$$

From now on, we assume that we applied the transformation given by (3.2) to the visual data. Hence, we have the zero-mean multivariate GP: $z(s) \sim \mathcal{GP}(0, \mathcal{K}(s, s'))$, which consists of multiple scalar GPs that are independent of each other.

3.3.1 The ρ-th random field

In this subsection, we only consider the ρ-th random field (visual feature). Other scalar random fields can be treated in the same way. The collection of n realized values of the ρ-th random field is denoted by $z_\rho := (z_\rho^{[1]}, \cdots, z_\rho^{[n]})^T \in \mathbb{R}^n$, where $z_\rho^{[i]} := z_\rho(s^{[i]})$ is the i-th realization of the ρ-th random field and $s^{[i]} = (s_1^{[i]}, s_2^{[i]}) \in \mathcal{S} \subset \mathbb{R}^2$ is the associated position where the realization occurs. We then have $z_\rho(s) \sim \mathcal{N}(0, \Sigma_\rho)$, where $\Sigma_\rho \in \mathbb{R}^{n \times n}$ is the covariance matrix. The i, j-th element of Σ_ρ is defined as $\Sigma_\rho^{[ij]} = \mathbb{C}\mathrm{ov}\,(z_\rho^{[i]}, z_\rho^{[j]})$. In this chapter, we consider the squared exponential covariance function [85] defined as

$$\Sigma_\rho^{[ij]} = \sigma_{f,\rho}^2 \exp\left(-\frac{1}{2}\sum_{\ell=1}^{2}\frac{(s_\ell^{[i]} - s_\ell^{[j]})^2}{\sigma_{\ell,\rho}^2}\right). \tag{3.3}$$

In general, the mean and the covariance functions of a GP can be estimated *a priori* by maximizing the likelihood function [114].

The prior distribution of z_ρ is given by $\mathcal{N}(0, \Sigma_\rho)$. A noise corrupted measurement $\tilde{z}_\rho^{[i]}$ at its corresponding location $s^{[i]}$ is defined as follows.

$$\tilde{z}_\rho^{[i]} = z_\rho^{[i]} + \epsilon_\rho^{[i]}, \tag{3.4}$$

where the measurement errors $\{\epsilon_\rho^{[i]}\}$ are assumed to be an independent and identically distributed (i.i.d.) Gaussian white noise, i.e., $\epsilon_\rho^{[i]} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_{\epsilon,\rho}^2)$. Thus, we have that

$$\tilde{z}_\rho \sim \mathcal{N}(0, R_\rho),$$

where $R_\rho = \left(\Sigma_\rho + \sigma_{\epsilon,\rho}^2 I\right)$. The log-likelihood function is defined by

$$L_{\theta,\rho} = -\frac{1}{2}\tilde{z}_\rho^T R_\rho \tilde{z}_\rho - \frac{1}{2}\log|R|_\rho - \frac{n}{2}\log 2\pi, \tag{3.5}$$

where n is the size of $\tilde{z}_\rho$.

The hyperparameter vector of the ρ-th random field is defined as $\theta_\rho = (\sigma_{f,\rho}, \sigma_{\epsilon,\rho}, \sigma_{1,\rho}, \sigma_{2,\rho}) \in \mathbb{R}_{>0}^4$. Using the likelihood function in (3.5) the hyperparameter vector can be computed by the ML estimator

$$\bar{\theta}_\rho = \arg\max_\theta L_{\theta,\rho}, \tag{3.6}$$

which will be plugged in prediction as in an empirical Bayes way.

All parameters are learned simultaneously. If no prior information is given, then the maximum a posteriori probability (MAP) estimator is equal to the ML estimator [114].

In a GP, every finite collection of random variables has a multivariate normal distribution. Consider a realized value of the ρ-th random field $z_\rho^\star$ being taken from the associated location $s^\star$. The probability distribution $\mathbb{P}(z_\rho^\star | s^\star, s, \tilde{z}_\rho)$ is a normal distribution with the following mean and variance.

$$\mu_\rho(s^\star) = C_\rho^T R_\rho^{-1} \tilde{z}_\rho, \quad \sigma_\rho^2(s^\star) = \sigma_{f,\rho}^2 - C_\rho^T R_\rho^{-1} C_\rho, \tag{3.7}$$

where the covariance $C_\rho := \mathbb{C}\mathrm{ov}\left(z^\star_\rho, z_\rho\right) \in \mathbb{R}^{1\times n}$ is defined similar to (3.3).

In order to estimate location $s^\star$, using the MAP estimator, we need to compute $\mathbb{P}(s^\star|\tilde{z}^\star_\rho, s, \tilde{z}_\rho)$, where the noisy observation $\tilde{z}^\star_\rho$ is the summation of the realized values of the random field $z^\star_\rho$ and a noise process.

$$\mathbb{P}(s^\star|\tilde{z}^\star_\rho, s, \tilde{z}_\rho) = \frac{\mathbb{P}(\tilde{z}^\star_\rho|s^\star, s, \tilde{z}_\rho)\,\mathbb{P}(s^\star|s, \tilde{z}_\rho)}{\mathbb{P}(\tilde{z}^\star_\rho|s, \tilde{z}_\rho)}. \tag{3.8}$$

A MAP estimator given the collection of observations $\tilde{z}_\rho$ is a mode of the posterior distribution.

$$\bar{s}^\star_\rho = \arg\max_{s^\star\in\mathcal{S}} \mathbb{P}(s^\star|\tilde{z}^\star_\rho, s, \tilde{z}_\rho). \tag{3.9}$$

If $\mathbb{P}(s^\star|s, \tilde{z}_\rho)$ and $\mathbb{P}(\tilde{z}^\star_\rho|s, \tilde{z}_\rho)$ are uniform probabilities, then the MAP estimator is equal to the ML estimator, given by

$$\bar{s}^\star_\rho = \arg\max_{s^\star\in\mathcal{S}} L_\rho(s^\star), \tag{3.10}$$

where the ρ-th log-likelihood function, i.e., $L_\rho(s^\star)$, is defined as follows.

$$L_\rho(s^\star) = \frac{-1}{2}\left(\frac{|\tilde{z}^\star_\rho - \mu_\rho(s^\star)|^2}{\sigma^2_{\epsilon,\rho} + \sigma^2_\rho(s^\star)} + \log\left(\sigma^2_{\epsilon,\rho} + \sigma^2_\rho(s^\star)\right) + \log 2\pi\right). \tag{3.11}$$

3.4 Localization and feature selection

Let Ω be the collection of indices that are associated to the multiple scalar random fields (of the multivariate GP). Provided that all scalar random fields (of the multivariate GP) are independent of each other, we then obtain a computationally efficient ML estimate of the location given the observations of all scalar random fields $\{\tilde{z}_\rho | \rho \in \Omega\}$ as follows.

$$\bar{s}^\star_\Omega = \arg\max_{s^\star \in \mathcal{S}} \sum_{\rho \in \Omega} L_\rho(s^\star), \tag{3.12}$$

where $L_\rho(s^\star)$ is the ρ-th log-likelihood function as given in (3.11).

In this chapter, a hold-out cross-validation technique [52] is used for the model selection. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. To this end, we divide the dataset into two segments: one used to learn or train the GP model and the other used to validate the model.

The RMSE is used to measure the performance of GP models. It is defined by the following equation.

$$\mathrm{RMSE}(\Omega) = \sqrt{\frac{1}{n_c} \sum_{i=1}^{n_c} \left\| s_c^{[i]} - \bar{s}^\star_\Omega \right\|^2}, \tag{3.13}$$

where $\| \cdot \|$ is the Euclidean norm of a vector. In the case that $\Omega = \emptyset$, we define the following.

$$\mathrm{RMSE}(\emptyset) = \sqrt{\frac{1}{n_c} \sum_{i=1}^{n_c} \left\| s_c^{[i]} - \mathrm{median}(s_c) \right\|^2},$$

where median$(\cdot)$ is the median of a random vector. Assume that $\Omega_m = \{1, \cdots, m\}$ is the set of all features. Dupuis et al. [23] reported that the backward sequential selection outperform the forward sequential selection. Thus, we use a backward sequential selection algorithm as

Algorithm 1 Learning maps from a sparse set of panoramic images observed in known locations

```
Input:  (1) training dataset includes a set of panoramic images captured from
        known spatial sites,
Output: (1) a linear transformation from image properties to uncorrelated
        visual features,
        (2) the estimated hyperparameter, the estimated mean and the esti-
        mated variance function of each independent visual feature,
1: extract image properties $y^{[i]}$ in the available learning dataset using (3.1)
2: use SVD to make a set of uncorrelated visual features $z^{[i]}$ using (3.2)
3: for each independent visual feature estimate hyperparameters using (3.6)
4: compute the mean function and variance function for each of independent features
   using (3.7)
5: choose optimal subset of visual features using (3.15) to eliminate some of the visual
   features that are worthless for the localization goal.
```

follows.

$$\Omega_{\ell-1} = \Omega_{\ell} - \arg\min_{\rho\in\Omega_{\ell}} \mathrm{RMSE}(\Omega_{\ell} - \rho), \forall \ell \in \Omega_m, \tag{3.14}$$

where $\Omega_{\ell} - \rho = \{p | p \in \Omega_{\ell}, p \neq \rho\}$.

Finally a subset of features is selected as follows.

$$\Omega_{opt} = \arg \min_{\Omega=\Omega_1,\cdots,\Omega_m} \mathrm{RMSE}(\Omega). \tag{3.15}$$

The optimum subset Ω_{opt} has the minimum RMSE among $\{\Omega_1, \cdots, \Omega_m\}$. The mapping and matching steps of the proposed approaches in this chapter are summarized in Algorithm 1 and Algorithm 2, respectively.

Algorithm 2 Localization predictive inference using learned map of visual features

```
Input:  (1) a linear transformation from image properties to uncorrelated
        visual features,
        (2) the estimated hyperparameter and the estimated mean and variance
        function of selected visual features,
Output: (1) position of newly captured images.

capture new images and obtain image properties $y^{\star}$ using (3.1)
compute the selected visual features $z^{\star}$ using (3.2)
compute the likelihood function of selected features $\rho \in \Omega_{opt}$ over possible sampling
  positions using (3.11)
determine the estimated position $\bar{s}^{\star}_{\Omega_{opt}}$ using (3.12)
```

3.5 Experimental Results

In this section, we demonstrate the effectiveness of the proposed localization algorithms using experiments. We report results on two different datasets of Case 1 and Case 2, obtained from two experimental setups.

3.5.1 Experimental setups

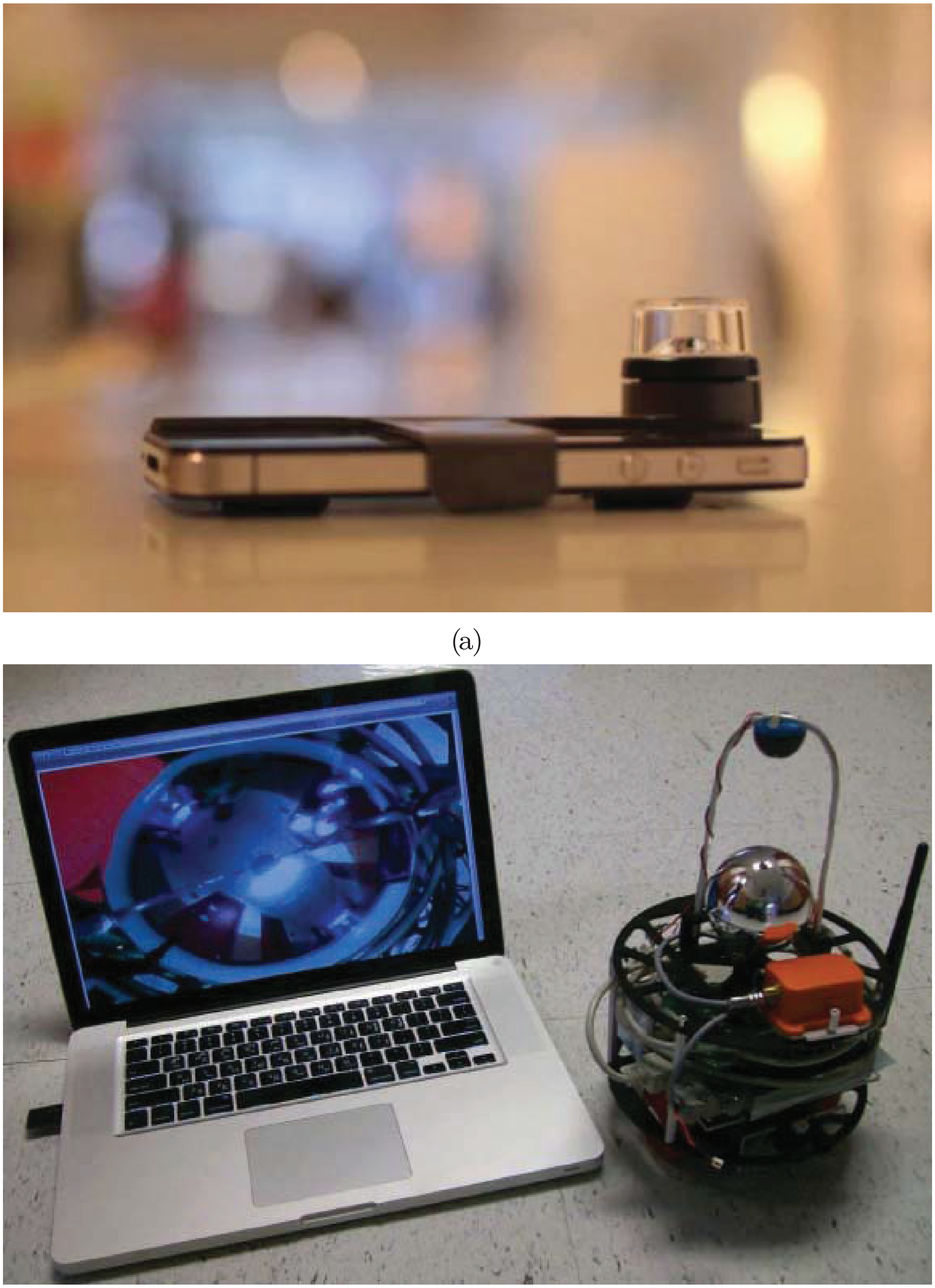
In Case 1, the Kogeto panorama lens was used to capture 360-degree images. In total, 207 pairs of exact sampling positions were recorded manually and corresponding captured panoramic images on a regular lattice (23×9 feet^2) were collected.

In Case 2, a vision-based robotic sensor was built by the authors to validate a proof of concept for the proposed methods. The wireless mobile robot is equipped with two motorized wheels, a micro-controller, a 360 degree omnidirectional camera, a wireless receiver, and a transmitter. The omnidirectional camera was built from a cheap camera (Ai-Ball Treck®) and a spherical mirror. The 360 degree images of the environment around the robot produced by the omnidirectional camera were steamed via 802.11 b/g Wi-Fi (see Fig. 3.2). In Case 2, we collected 825 pairs of noisy sampling positions recorded automatically by image processing software (using an exterior camera), and corresponding captured panoramic images. The noise power of the image processing software is 1 square foot, and the samples were taken on a 10×26 feet^2 surface.

Figs. 3.2-(a) and 3.2-(b) show two different hardware setups for Case 1 and Case 2, respectively. The datasets are divided into 80% *learning*, 10% *cross-validation* and 10% *testing* subsets. The learning subset is used to estimate hyperparameters, mean functions, and variance functions of GP models for the scalar fields. The cross-validation subset is used to select the best features in order to minimize the localization estimation error. After the training, the testing subset is used to evaluate the algorithm performance.

3.5.2 Learning of GP models in an empirical Bayes approach

As illustrative examples, we apply the proposed algorithm on both dataset with 32 computed visual features. Figs. 3.3 and 3.4 show the estimated hyperparameters ($k \in \Omega_{32}$) for Case 1 and Case 2, respectively. The variance of the random field σ_f, the spatial bandwidths β_1 and β_2, and the noise variance σ_ϵ are estimated for each independent feature, individually. Thus, $32 \times 4 = 128$ hyperparameters have been estimated in total for each experimental setup. In Case 1, the predictive mean and variance of each visual features $\{z_k | k \in \Omega_{32}\}$ are shown in Figs. 3.5 and 3.6, respectively. The predictive mean and variance for Case 2 are shown similarly in Figs. 3.7 and 3.8, respectively. The fixed running time using Matlab R2009b on a PC (3.2 GHz Intel i7 Processor) is summarized in Table 3.1, for Case 2 (similar results are observed for Case 1). Localization for each test point takes 0.2 sec which is fast enough for a real time implementation.



(a)

(b)

Figure 3.2: (a) Dot iPhone Panorama lens is used for Case 1. (b) A vision-based robot was built equipped with a 360 degree omnidirectional camera for Case 2.

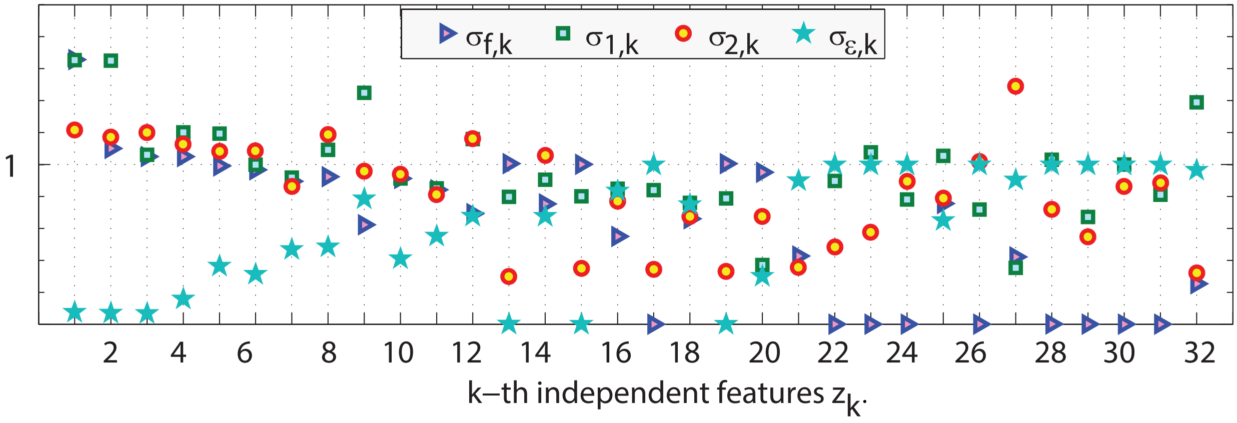


Figure 3.3: Estimated hyperparameters for each individual scalar field for Case 1.

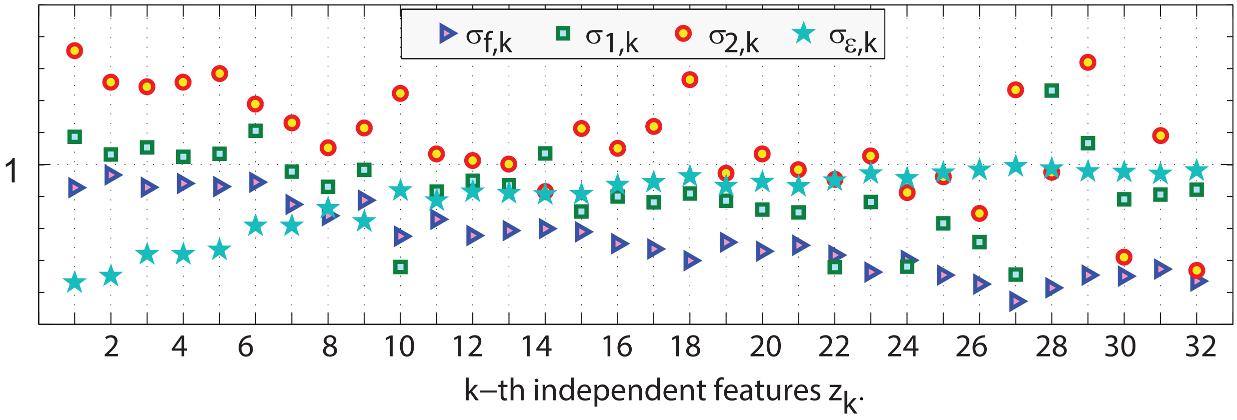


Figure 3.4: Estimated hyperparameters for each individual scalar field for Case 2.

3.5.3 Comparison study

We compare the performance of the proposed approach under different image features as summarized in Table 3.2. In this comparison study three different types of appearance-based features such as FFT [110], histogram [34], and wavelet [7] are considered. Additionally, the comparison between the proposed cross-validation (CV) method and the traditional PCA in feature elimination is investigated. Finally, we compared the result of localization methods in terms of different number of features.

Table 3.2 shows the dataset (column 1), the appearance-based features type (column 2), the total number of features (column 3), the optimum number of features determined by the

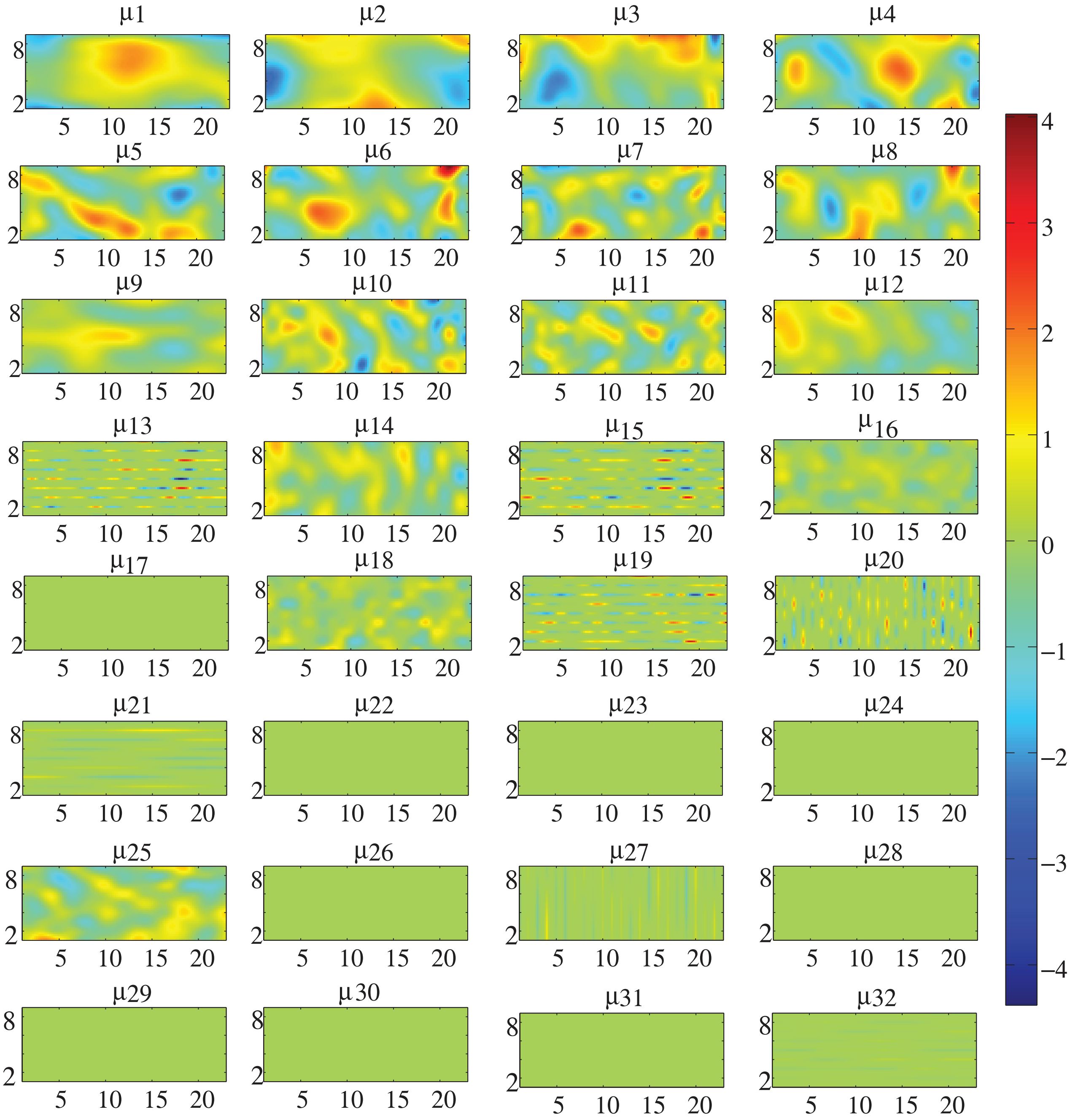


Figure 3.5: Predictive mean function map for each individual feature z_k for Case 1.

proposed approach (column 4), the localization RMSE obtained using the total number of features (column 5), the sixteen features selected by PCA (column 6), the sixteen features selected by CV (column 7), and the optimum number of features (column 8).

Note that sampling positions of Case 1 were recorded exactly while those of Case 2

Table 3.1: Computational time

Procedure	**Number of observations**	**Time (sec)**
Pre-processing	825	35
Feature extraction	825	2
Learning hyperparameters	659	286
Learning map and variances	659	51
Feature selection	83	21
Localization	83	10

Table 3.2: The localization performance comparison

		number of features		RMSE using subset of features			
	feature type	total	optimum	All	PCA 16	CV 16	CV opt
Case 1	FFT	8	7	4.98	-	-	0.63
	FFT	16	13	4.98	-	-	0.53
	FFT	32	10	4.98	4.98	0.24	0.22
	FFT	64	26	4.98	4.98	0.31	0.26
	Histogram	8	5	4.12	-	-	2.10
	Histogram	16	10	3.85	-	-	1.57
	Histogram	32	13	1.44	1.43	1.22	1.18
	Histogram	64	32	1.46	1.56	1.15	1.13
	Wavelet	78	32	6.69	4.43	4.03	2.49
Case 2	FFT	8	8	2.21	-	-	2.21
	FFT	16	11	2.00	-	-	1.58
	FFT	32	21	1.52	2.42	1.37	1.36
	FFT	64	21	1.96	2.54	1.24	1.19
	Histogram	8	5	2.88	-	-	2.38
	Histogram	16	9	2.60	-	-	1.77
	Histogram	32	19	2.03	1.54	1.59	1.47
	Histogram	64	32	2.08	2.23	1.53	1.40
	Wavelet	78	27	5.96	7.94	3.82	3.52

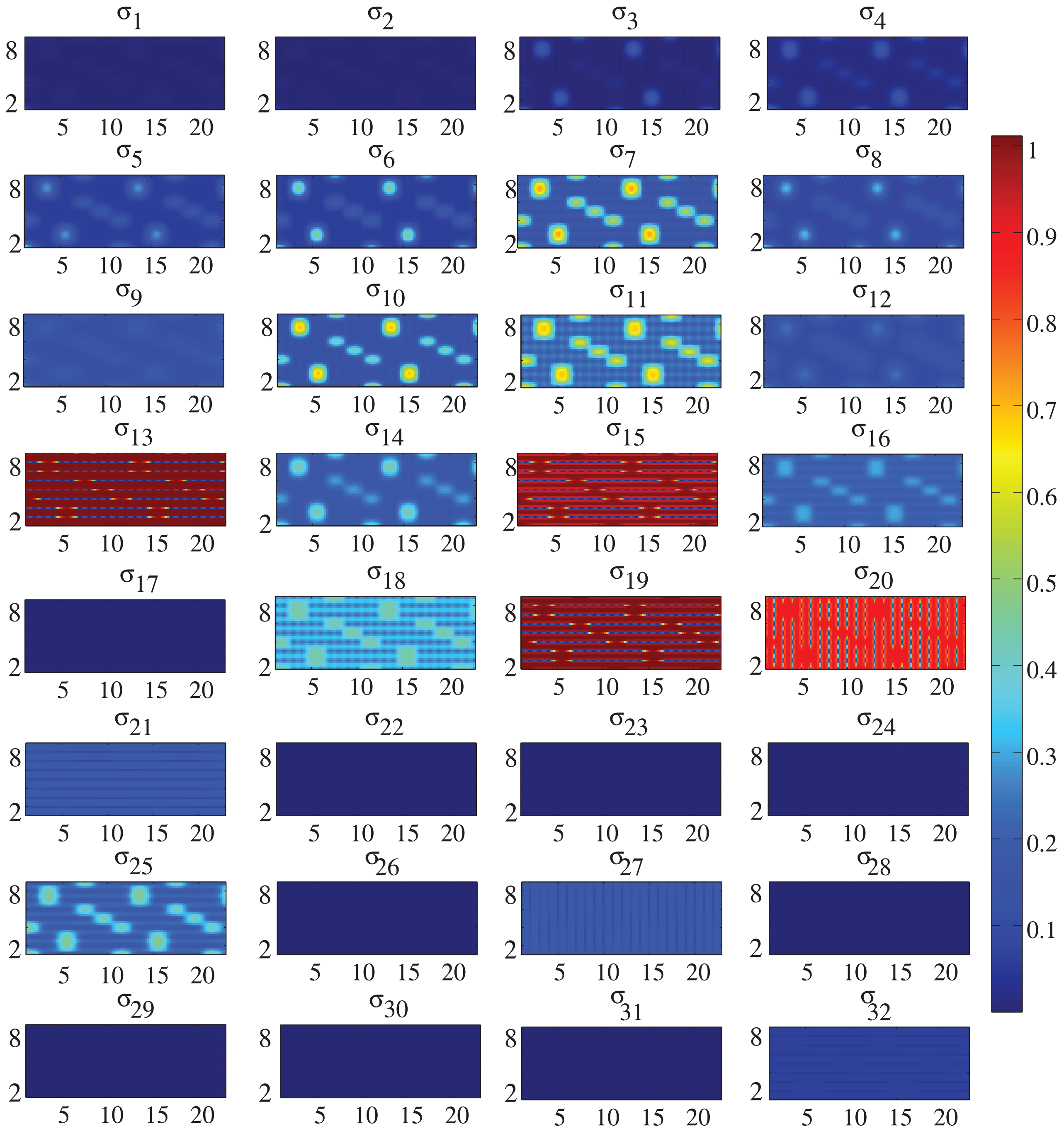


Figure 3.6: Prediction variance map for each individual feature z_k for Case 1.

were noisy. As expected, the results of Case 1 outperform those of Case 2. This could be due to the fact that noisy sampling positions lead to poorly estimated hyperparameters and predictions of the GP. Furthermore, the FFT appearance-based features show more vulnerability against sampling position noise. The RMSE results of FFT-based localization

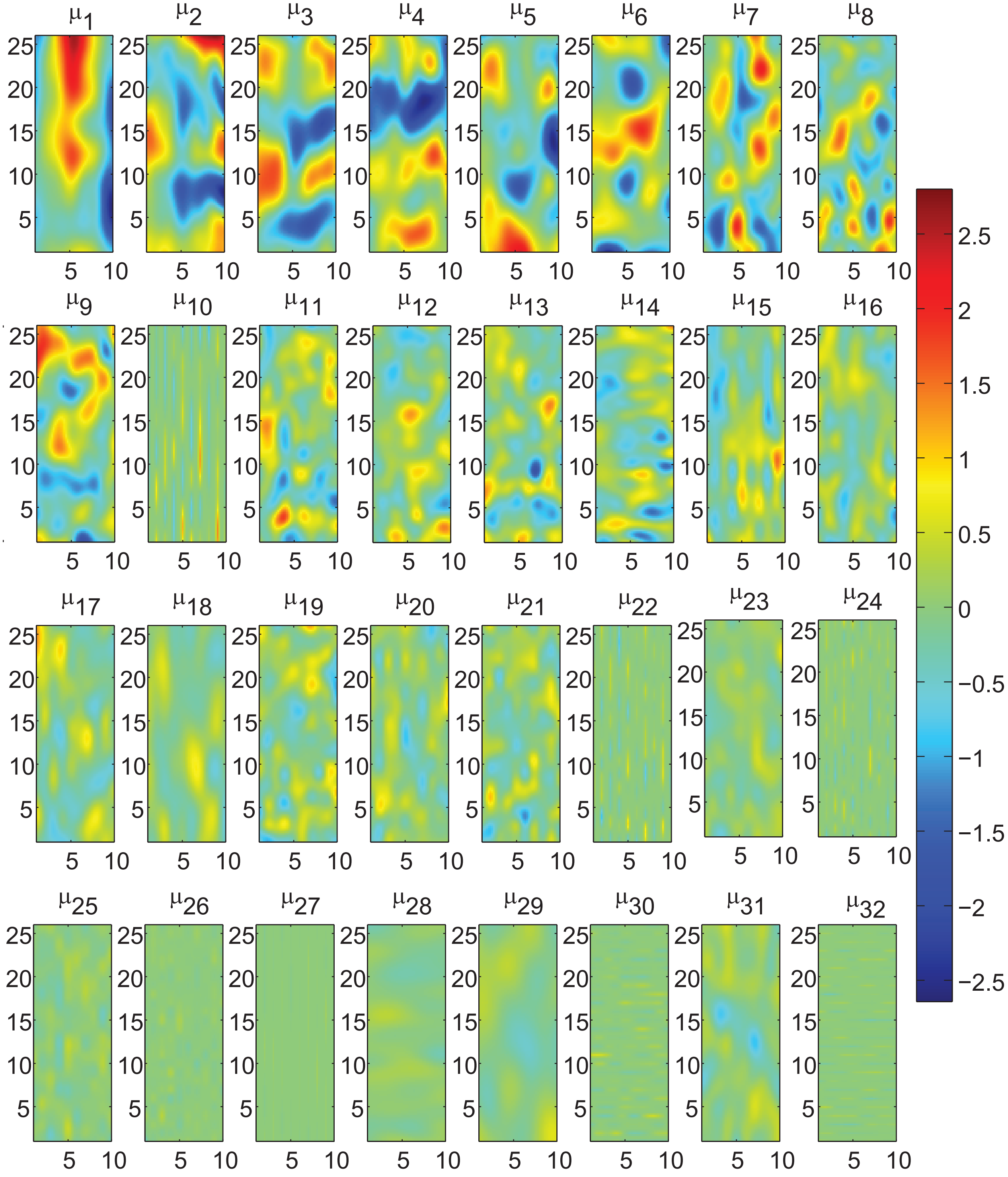


Figure 3.7: Predictive mean function map for each individual feature z_k for Case 2.

are slightly better than those of histogram-based localization and much better than those of wavelet-based localization [7]. Besides the type and total number of features, the CV has

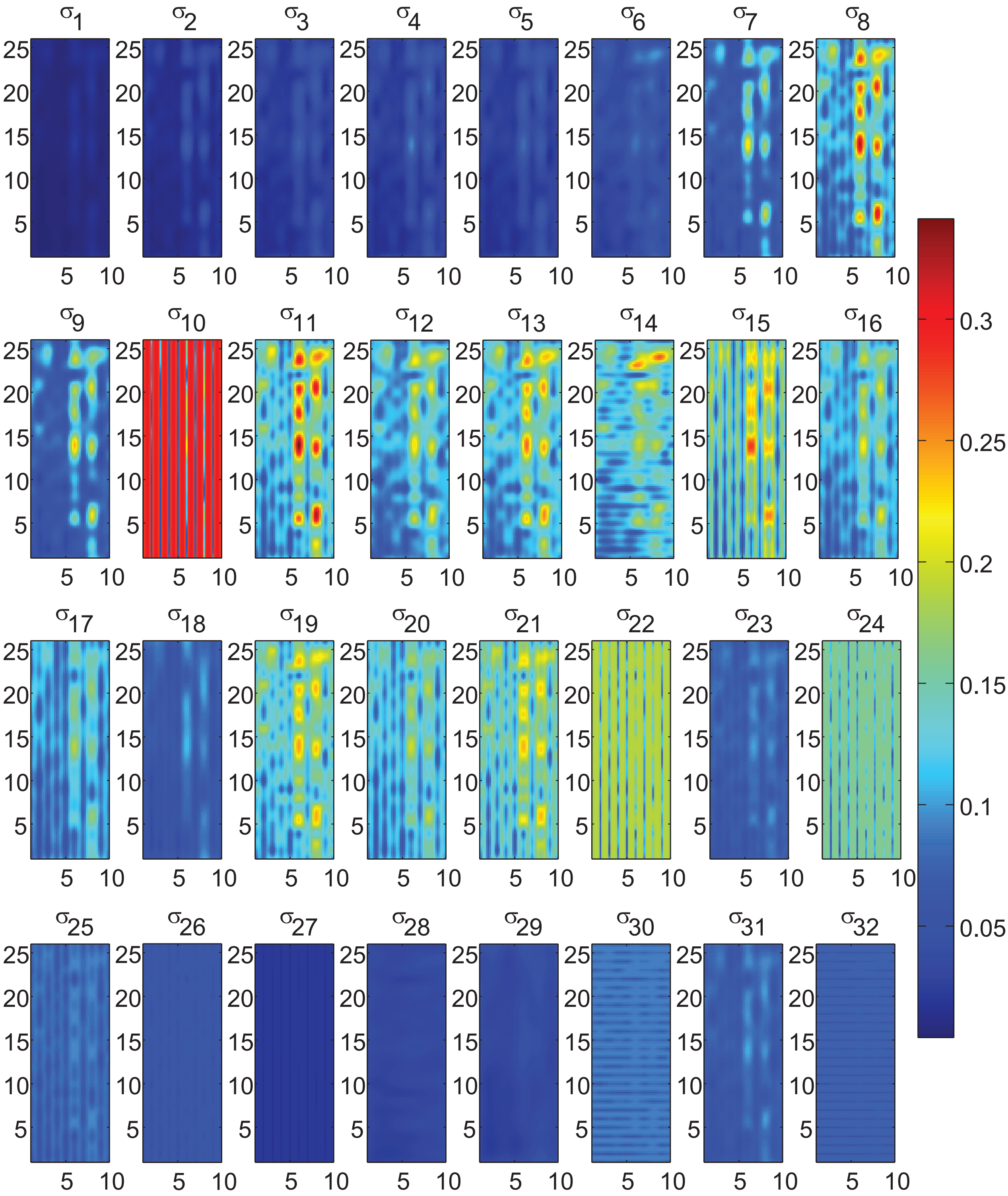


Figure 3.8: Prediction variance map for each individual feature z_k for Case 2.

a better performance with respect to PCA. In general, increased cardinality of the original feature set leads to a better performance in the 5th column possibly due to the increased information. Similar trends are observed in the 7th and the 8th columns of Table 3.2.

Most importantly, we conclude that eliminating some of the features helps to improve the localization error. Therefore, the RMSE of the proposed approach in this chapter is much lower than other methods.

3.6 Conclusion and future works

This chapter has presented a novel approach to use vision data for the mobile robot localization. The predictive statistics of vision data is learned in advance and used in order to estimate the position of mobile robot, equipped just with an omnidirectional camera in a GPS-denied environment. The multivariate GP model with unknown hyperparameters is used to model a collection of selected visual features. Experimental study shows excellent positioning results within a reasonable computational time under two different experimental setups with and without uncertainty in sampling positions.

A key limitation of the current approach arises from the fact that, after the initial training phase, learning is discontinued. In particular, if the environment changes, it is desirable that the localization routines adapt to the changes in the environment. Thus, a future research direction is develop a localization scheme that is adaptive to changes in the environment.

Chapter 4

Bayesian Prediction with Uncertain Localization

Abstract: In this chapter, we consider the problem of predicting a spatio-temporal random field using sequential noisy observations collected by robotic sensors. The spatio-temporal field of interest is modeled by a sum of a time-varying mean function and a Gaussian Markov random field (GMRF) with unknown hyperparameters. We first derive the exact Bayesian solution to the problem of computing the predictive inference of the random field, taking into account observations, uncertain hyperparameters, measurement noise, and uncertain localization in a fully Bayesian point of view. We show that the exact solution for uncertain localization is not scalable as the number of observations increases. To cope with this exponentially increasing complexity and to be usable for mobile sensor networks with limited resources, we propose a scalable approximation with a controllable tradeoff between approximation error and complexity to the exact solution. The effectiveness of the proposed algorithms is demonstrated by simulation and experimental results.

4.1 Introduction

In recent years, there has been an increasing exploitation of mobile robotic sensors in environmental monitoring [10,12,62,64,117]. Gaussian processes (or Gaussian random fields) defined by mean and covariance functions over a continuum space [14,84] have been frequently used for mobile sensor networks to statistically model physical phenomena such as harmful algal blooms, pH, and temperature, e.g., [30,57,119].

The significant computational complexity in Gaussian process regression due to the growing number of observations (and hence the size of covariance matrix) has been tackled in different ways. In [119], the conditions under which near-optimal prediction can be achieved is analyzed, using truncated observations when the covariance function is known *a priori*. In [120], a new efficient and scalable inference algorithm for a class of Gaussian processes that builds on a Gaussian Markov random field (GMRF) is developed for known hyperparameters. On the other hand, unknown hyperparameters in the covariance function can be estimated by a maximum likelihood (ML) estimator or a maximum *a posteriori* (MAP) estimator and then can be used for the prediction [114]. However, the point estimate itself needs to be identified using certain amount of measurements and it does not fully incorporate the uncertainty in the estimated hyperparameters into the prediction in a Bayesian perspective.

The advantage of a fully Bayesian approach is the capability of incorporating various uncertainties in the model parameters and measurement processes in the prediction [3]. However, the solution often requires an approximation technique such as Markov Chain Monte Carlo (MCMC), Laplace approximation, or variational Bayes methods, which still requires a high level of computational complexity [3]. In [115,117], a sequential Bayesian prediction algorithm and its distributed version are designed to deal with uncertain bandwidths by using a compactly supported kernel and selecting a subset of collected measurements. Sequential fully Bayesian prediction algorithms for a GMRF with unknown hyperparameters are reported in [116].

Inexpensive wireless/mobile sensor networks [17, 79] are widespread at the cost of precision in localization. Due to their growing usage, there are many practical opportunities where continuously sampled measurements need to be fused for sensors with *localization uncertainty*. Theoretically-correct yet efficient (or scalable) inference algorithms need to be developed to meet such demands.

Related works involving uncertain localization in our context are as follows. Gaussian process regression has been used in building map and localization in many practical applications. In [7], Gaussian process regression is used to model geo-referenced sensor measurements (obtained from a camera). After training, an ML estimator is used to find the best match for the locations of newly sampled measurements. However, the training step should be done in advance for a new environment using a set of data including noisy measurements and the exact sampling positions [7]. In [51, 105], Gaussian process regression is used to implement simultaneous localization and mapping (SLAM) using a magnetic field and its feasibility is shown experimentally. In [77, 106], the problem of using laser range-finder data is investigated to probabilistically classify a robot's environment. They provide a continuous representation of the robot's surroundings by employing a Gaussian process. In [65], a Gaussian process model is presented for training on input points corrupted by independent and identically distributed (i.i.d.) Gaussian noise. To make the computation tractable a local linear expansion is used about each sampling position for the mean function. The work in [65] assumes that all hyperparameters are trained offline *a priori*. In [44], the problem of Gaussian process regression with uncertain localization and known hyperparameters is formulated and solved. A key limitation of such a approach with fixed hiperparameters arises from the fact that, after the initial training phase, learning is discontinued. In general, if the environment changes, it is desirable that the localization algorithm adapts to the changes on the fly. A fully Bayesian approach that treats hyperparameters as random variables can address this issue with increased computational complexity.

The novelty of our work as compared to the previous ones in [115–117] is to fully consider the uncertainty on the sampling positions along with other uncertainties such as hyperparameters, observation noise etc., in a *fully Bayesian* manner. To the best of our knowledge, fully Bayesian prediction algorithms for spatio-temporal random fields that can take into account uncertain localization is scant to date. Hence, this chapter aims to develop such inference algorithms for robotic sensor networks in practical situations. With continuous improvement in computation power in embedded systems, it is very important to prepare theoretically-correct, and flexible fully Bayesian approach to cope with such practical problems.

The contributions of the chapter are as follows. First, we model a physical spatio-temporal random field as a GMRF with uncertain hyperparameters and formulate the prediction problems with and without localization uncertainty. Next, we derive the exact Bayesian solution to the problem of computing the predictive inference of the random field, taking into account uncertain hyperparameters, measurement noise, and uncertain localization in a fully Bayesian point of view. We show that the exact solution for uncertain localization is not scalable as the number of observations increases. To cope with this increasing complexity, we propose a scalable approximation with a controllable tradeoff between approximation error and complexity to the exact solution. The effectiveness of the proposed algorithms is demonstrated by simulation and experimental results. In particular, experimental results are illustrated in both static and dynamical environments.

A preliminary version of this work without the proofs of theorems and experimental results has been reported in [41].

The chapter is organized as follows. In Section 4.2, we explain how a GMRF can be viewed as a sparse and discretized version of a Gaussian process. In Sections 4.3 and 4.4, we introduce a spatio-temporal field model based on a GMRF and the mobile sensor network. In Section 4.5.1, we present a fully Bayesian inference approach to estimate the spatio-temporal field. The Bayesian prediction algorithm is extended for uncertain sampling positions in Section 4.5.2. The complexity of the proposed algorithms is discussed in Section 4.5.3. We demonstrate the effectiveness through a simulation study in Section 4.6. Finally, we evaluate our approach on a real experimental setup in Section 4.7.

4.2 Gaussian Process and Gaussian Markov Random fields

Recently, there are efforts to fit a computationally efficient GMRF on a discrete lattice to a Gaussian random field on a continuum space [15,35,90,116]. It has been demonstrated that GMRFs with small neighborhoods can approximate Gaussian fields surprisingly well [90]. This approximated GMRF and its regression are very attractive for the resource-constrained mobile sensor networks due to its computational efficiency and scalability [60] as compared to the standard Gaussian process and its regression. Fast kriging of large data sets by using a GMRF as an approximation of a Gaussian random field has been proposed in [35].

We now briefly review a GMRF as a discretized Gaussian process on a lattice. Consider a zero-mean Gaussian process: $z(s) \sim \mathcal{GP}(0, \mathcal{K}(s, s'))$, where $\mathcal{K}$ is the covariance function defined in a continuum space $\mathcal{S}$. We discretize the compact domain $\mathcal{S} := [0 \ \max(s_1)] \times [0 \ \max(s_2)]$ into n spatial sites $\mathcal{Q} := \left\{\mathring{s}^{(1)}, \cdots, \mathring{s}^{(n)}\right\} \subset \mathbb{R}^2$, where $n = h\max(s_1) \times h\max(s_2)$. h will be chosen such that $n \in \mathbb{Z}_{>0}$. Note that $n \to \infty$ as $h \to \infty$. The collection of realized values of the random field in $\mathcal{Q}$ is denoted by $\mathbf{z} := (z^{(1)}, \cdots, z^{(n)})^T \in \mathbb{R}^n$, where $z^{(i)} := z(\mathring{s}^{(i)})$.

The prior distribution of $\mathbf{z}$ is given by $\mathbf{z} \sim \mathcal{N}(0, \mathbf{K})$, and so we have

$$\mathbb{P}(\mathbf{z}) \propto \exp\left(-\frac{1}{2}\mathbf{z}^T\mathbf{K}^{-1}\mathbf{z}\right), \tag{4.1}$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is the covariance matrix. The i, j-th element of $\mathbf{K}$ is defined as $(\mathbf{K})_{ij} = \mathbb{C}\mathrm{ov}(z^{(i)}, z^{(j)}) = \mathcal{K}(z^{(i)}, z^{(j)})$. The prior distribution of $\mathbf{z}$ can be written by a precision matrix $\mathbf{Q}_0 = \mathbf{K}^{-1}$, i.e., $\mathbf{z} \sim \mathcal{N}(0, \mathbf{Q}_0^{-1})$. This can be viewed as a discretized version of the Gaussian process (or a GMRF) with a precision matrix $\mathbf{Q}_0$ on $\mathcal{Q}$. Note that $\mathbf{Q}_0$ of this GMRF is not sparse. However, a sparse version of $\mathbf{Q}_0$, i.e., $\hat{\mathbf{Q}}_0$ with local neighborhood that can represent the original Gaussian process can be found, for example, making $\hat{\mathbf{Q}}_0$ close to $\mathbf{Q}_0$ in some norm [15,35,90]. This approximate GMRF will be computationally efficient due to the sparsity of $\hat{\mathbf{Q}}_0$. For our main problems, we will use a GMRF with a sparse precision matrix that represents a Gaussian process precisely (see Section 4.4).

In this section, we assume that we take N noisy measurements $\mathbf{y} = (y^{[1]}, \cdots, y^{[N]})^T \in \mathbb{R}^N$ from corresponding sampling locations $\mathbf{s} = (s^{[1]T}, \cdots, s^{[N]T})^T \in \mathcal{S}^N$. Here, the measurement model is given by

$$y^{[i]} := y(s^{[i]}) = z(s^{[i]}) + \epsilon^{[i]}, \forall i = 1, \cdots, N \tag{4.2}$$

where $\epsilon^{[i]} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, {\sigma_\epsilon}^2)$ is the measurement noise and is assumed to be independent and identically distributed (i.i.d.).

Using Gaussian process regression, the posterior distribution for $\mathbf{z} \in \mathbb{R}^n$ is given by

$$\mathbf{z}|\mathbf{s}, \mathbf{y} \sim \mathcal{N}(\mu, \Sigma). \tag{4.3}$$

The predictive mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ can by obtained by

$$\mu = \mathbf{k}^T \mathbf{C}^{-1} \mathbf{y}, \quad \Sigma = \mathbf{K} - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k}, \tag{4.4}$$

where the covariance matrices are defined as $\mathbf{k} := \mathbb{C}\text{ov}(\mathbf{y}, \mathbf{z}) \in \mathbb{R}^{N \times n}$, $\mathbf{C} := \mathbb{C}\text{ov}(\mathbf{y}, \mathbf{y}) \in \mathbb{R}^{N \times N}$, and $\mathbf{K} := \mathbb{C}\text{ov}(\mathbf{z}, \mathbf{z}) \in \mathbb{R}^{n \times n}$.

In practice, resource-constrained robotic sensors are prone to large uncertainty in localization. Most previous works on Gaussian process regression for mobile sensor networks [8, 9, 14, 57] have assumed that the exact sampling positions are available, which is not practical in real situations. Localization uncertainty effect is potentially significant. For example, Fig. 2.1 shows the effect of noisy sampling positions on the results of Gaussian process regression. Note that adding noise to the sampling positions considerably increase the empirical RMS error, shown in the third row of Fig. 2.1. This motivates our work in this chapter.

The pose of a robot can be estimated by fusing different sensory information producing its estimate and estimation error statistics. In general, Bayes filters such as extended and unscented Kalman filters [47, 96], particle filters [70], and Monte Carlo techniques [98] are used for such localization. Nevertheless, obtaining precise localization of robotic networks under limited resources is very challenging [37,38].

Throughout the chapter, we assume that the positions of robotic sensors are estimated by a standard technique that produces the position estimates along with estimation error statistics. A discussion regarding specific localization technique is standard and is beyond the scope of this chapter.

Having had the aforementioned assumption, from a localization algorithm, the prior distribution for sampling location $s^{[i]}$ is given as $\mathbb{P}(s^{[i]}|\tilde{s}^{[i]})$, possibly with a compact support in $\mathcal{S}$. Then the predictive distribution of $\mathbf{z}$ given the measured locations $\tilde{\mathbf{s}} = (\tilde{s}^{[1]T}, \cdots, \tilde{s}^{[N]T})^T$ is thus given by

$$\mathbb{P}(\mathbf{z}|\tilde{\mathbf{s}}, \mathbf{y}) = \int_{\mathbf{s} \in \mathcal{S}^N} \mathbb{P}(\mathbf{z}|\mathbf{s}, \mathbf{y}) \, \mathbb{P}(\mathbf{s}|\tilde{\mathbf{s}}, \mathbf{y}) d\mathbf{s}, \tag{4.5}$$

where $\mathbb{P}(\mathbf{z}|\mathbf{s}, \mathbf{y})$ can be obtained in (4.3). However, the predictive distribution in (4.5) does not have a closed-form solution and needs to be computed either by MCMC methods or approximation techniques [43].

Now we consider a discretized version of the Gaussian process, i.e., (GMRF) with a precision matrix $\mathbf{Q}_0$ on $\mathcal{Q}$. Since the sampling points of Gaussian process regression are not necessarily on $\mathcal{Q}$, we use the nearest grid point of a given sampling point s in $\mathcal{S}$

$$q^{[i]} = \arg\min_{q \in \mathcal{Q}} \|s^{[i]} - q\|.$$

The sampling positions for the GMRF are then exactly on the lattice, i.e., $q^{[i]} \in \mathcal{Q}$. The posterior distribution of $\mathbf{z} \in \mathbb{R}^n$ on $\mathcal{Q}$ given by measurements in $\mathbf{y} \in \mathbb{R}^N$ and sampling positions in $\mathbf{q} = (q^{[1]T}, \cdots, q^{[N]T})^T \in \mathcal{Q}^N$, is then obtained by

$$\mathbf{z}|\mathbf{q}, \mathbf{y} \sim \mathcal{N}(\mathbf{Q}^{-1} b, \mathbf{Q}^{-1}), \tag{4.6}$$

where $\mathbf{Q} = \mathbf{Q}_0 + HP^{-1}H^T$, $b = HP^{-1}\mathbf{y}$, with $P = {\sigma_\epsilon}^2 I \in \mathbb{R}^{N \times N}$ and $H \in \mathbb{R}^{n \times N}$ defined as

$$(H)_{ij} = \begin{cases} 1, & \text{if } \mathring{s}^{(i)} = q^{[j]}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

We consider again localization uncertainty for this GMRF. Let the measured noisy location $\tilde{q}^{[i]}$ be the nearest grid point of the measured noisy sampling point $\tilde{s}^{[i]}$ of the Gaussian process. Now we obtain a set of discretized probabilities in $\mathcal{Q}$ induced by the continuous prior distribution defined in $\mathcal{S}$. The discrete prior distribution for the sampling location $q^{[i]}$ is given by

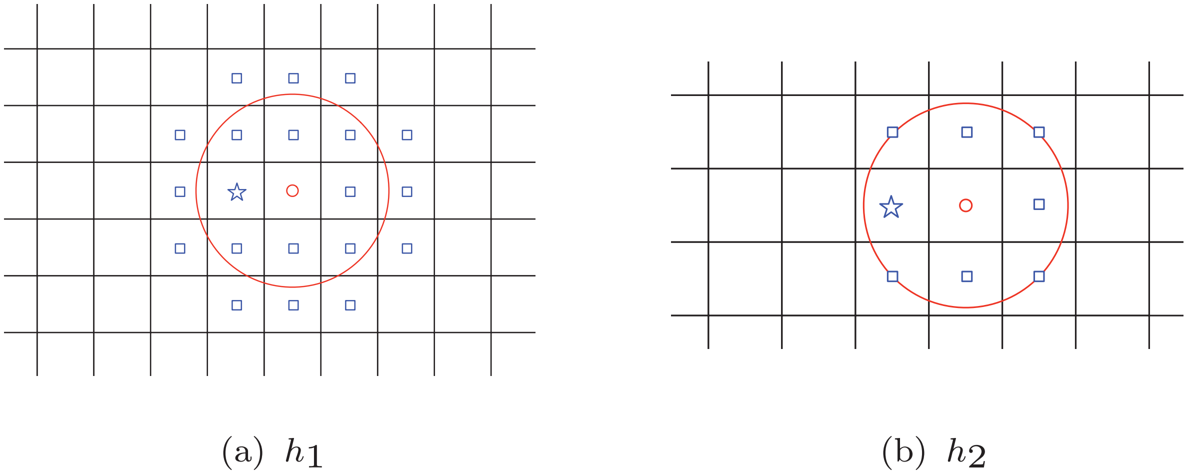
$$\mathbb{P}(q^{[i]} = \mathring{s}^{(j)} | \tilde{s}^{[i]}) = \int_{s \subset \mathcal{V}_j} \mathbb{P}(s|\tilde{s}^{[i]}) ds, \tag{4.8}$$

where $\mathbb{P}(s|\tilde{s})$ is the continuous prior as in Gaussian process regression and $\mathcal{V}_j$ is the Voronoi cell of the j-th grid point $\mathring{s}^{(j)}$ given by

$$\mathcal{V}_j := \{ s \in \mathcal{S} \,|\, \|s - \mathring{s}^{(j)}\| \leq \|s - \mathring{s}^{(i)}\|, \forall i \neq j \}.$$

The predictive distribution of $\mathbf{z}$ given $\mathbf{y}$ and $\tilde{\mathbf{q}}$ is thus given by

$$\mathbb{P}(\mathbf{z}|\tilde{\mathbf{q}}, \mathbf{y}) = \sum_{\mathbf{q} \in \mathcal{Q}^N} \mathbb{P}(\mathbf{z}|\mathbf{q}, \mathbf{y}) \, \mathbb{P}(\mathbf{q}|\tilde{\mathbf{q}}, \mathbf{y}), \tag{4.9}$$


(a) h_1

(b) h_2

Figure 4.1: Example of localization uncertainty for $s^{[i]}$. The measured sampling location $\tilde{q}^{[i]}$ is indicated in a small red circle which is the closest point in the discrete support to the measured sampling position in the continuous space. The small red circle along with the blue squares and the blue star show the possible locations of the true sampling point $q^{[i]}$ according to the prior distribution $\mathbb{P}(q^{[i]} = \mathring{s}^{(j)} | \tilde{q}^{[i]})$ with a compact support as shown in the big red circle. The blue star indicates $q^{[i]}$ which is the closest point in the discrete support to the true sampling position in the continuous space.

where $\mathbb{P}(\mathbf{z}|\mathbf{q}, \mathbf{y})$ can be obtained by (4.6) and the summation is over all possible locations in $\mathcal{Q}$. Fig. 4.1 shows two examples of using this approximation approach with $h_1 > h_2$ to convert a continuous space to a discrete one. When $h \to \infty$, $\tilde{q} \to \tilde{s}$ and the standard Gaussian process regression in a continuum space shall be recovered from the prediction using the GMRF in a discretized space.

4.3 Mobile Senor Networks

Suppose that the sampling time $t \in \mathbb{Z}_{>0}$ is discrete. Let $\mathbf{z}_t := (z_t^{(1)}, \cdots, z_t^{(n)})^T \in \mathbb{R}^n$ be the corresponding values of the scalar field at n special sites and time t.

Consider N spatially distributed mobile sensing agents indexed by $j \in \mathcal{J} := \{1, \cdots, N\}$ sampling at time $t \in \mathbb{Z}_{>0}$. At time t, agent j takes a noise corrupted measurement at its current location $s_t^{[j]} \approxeq \mathring{s}^{(i)} \in \mathcal{Q}$, i.e.,

$$y_t^{[j]} = z_t^{(i)} + \epsilon_t^{[j]}, \quad \epsilon_t^{[j]} \overset{i.i.d.}{\sim} \mathcal{N}(0, {\sigma_\epsilon}^2), \tag{4.10}$$

where the measurement errors $\{\epsilon_t^{[j]}\}$ are assumed to be i.i.d. The measurement noise level ${\sigma_\epsilon}^2 > 0$ is assumed to be known. For notational simplicity, we denote all agents' locations at time t by $\mathbf{q}_t = \left(q_t^{[1]T}, \cdots, q_t^{[N]T}\right)^T \in \mathcal{Q}^N$ and the observations made by all agents at time t by $\mathbf{y}_t = \left(y_t^{[1]}, \cdots, y_t^{[N]}\right)^T \in \mathbb{R}^N$. Furthermore, we denote the collection of agents' locations and the collective observations from time 1 to t by $\mathbf{q}_{1:t} = \left(\mathbf{q}_1^T, \cdots, \mathbf{q}_t^T\right)^T \in \mathcal{Q}^{Nt}$ and $\mathbf{y}_{1:t} = (\mathbf{y}_1, \cdots, \mathbf{y}_t)^T \in \mathbb{R}^{Nt}$, respectively. In addition, let us define $\mathbf{z}_t = (z_t^{(1)}, \cdots, z_t^{(n)})^T \in \mathbb{R}^n$ on $\mathcal{Q}$, and $\boldsymbol{\epsilon}_t = (\epsilon_t^{[1]}, \cdots, \epsilon_t^{[N]})^T \in \mathbb{R}^N$. We then have the following collective notation.

$$\mathbf{y}_t = H_t^T \mathbf{z}_t + \boldsymbol{\epsilon}_t, \tag{4.11}$$

where $H_\tau \in \mathbb{R}^{n \times N}$ is defined by

$$(H_\tau)_{ij} = \begin{cases} 1, & \text{if } \mathring{s}^{(i)} = q_\tau^{[j]}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

4.4 Spatio-Temporal Field Model

The value of the scalar field at $\mathring{s}^{(i)}$, $z_t^{(i)}$ is modeled by a sum of a time-varying mean function and a GMRF

$$z_t^{(i)} = \lambda_t^{(i)} + \eta_t^{(i)}, \forall i \in \{1, \cdots, n\}, \, t \in \mathbb{Z}_{>0}. \tag{4.13}$$

Here the mean function $\lambda_t^{(i)} : \mathcal{Q} \times \mathbb{Z}_{>0} \to \mathbb{R}$ is defined as

$$\lambda_t^{(i)} = f(\mathring{s}^{(i)})^T \beta_t, \tag{4.14}$$

where $f(\mathring{s}^{(i)}) = (f_1(\mathring{s}^{(i)}), \cdots, f_p(\mathring{s}^{(i)}))^T \in \mathbb{R}^p$ is a known regression function and $\beta_t = (\beta_t^{[1]}, \cdots, \beta_t^{[p]})^T \in \mathbb{R}^p$ is an unknown vector of regression coefficients. The time evolution of $\beta_t \in \mathbb{R}^p$ is modeled by a linear time-invariant system given by

$$\beta_{t+1} = A_t \beta_t + B_t \omega_t, \tag{4.15}$$

where $\omega_t \sim \mathcal{N}(0, W)$, $\beta_0 \sim \mathcal{N}\left(\mu_{\beta_0}, \Sigma_{\beta_0}\right)$, and A_t and B_t are known system parameters.

In addition, we consider a zero-mean GMRF [88] $\eta_t = \left(\eta_t^{(1)}, \cdots, \eta_t^{(n)}\right)^T \in \mathbb{R}^n$ whose covariance matrix is given by

$$\mathbb{E}(\eta_t \eta_k^T | \theta) = \mathbf{Q}_\theta^{-1} \delta(t - k), \tag{4.16}$$

where $\delta(\cdot)$ is the Kronecker delta defined by

$$\delta(k) = \begin{cases} 1, & k = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{4.17}$$

and the inverse covariance matrix (or precision matrix) $\mathbf{Q}_\theta \in \mathbb{R}^{n \times n}$ is a function of the hyperparameter vector θ.

There are different parameterizations of the GMRF (i.e., the precision matrix $\mathbf{Q}_\theta$) [88]. Our Bayesian approach does not depend on the choice of the parameterization for the precision matrix. However, for a concrete and useful exposition, we describe a specific parameterization used in this chapter. The precision matrix is parameterized with the full conditionals as follows.

Let η be a GMRF on a regular two-dimensional lattice. The associated Gaussian full conditional mean is

$$\mathbb{E}(\eta_t^{(i)}|\eta_t^{(-i)},\theta) = -\frac{1}{(\mathbf{Q}_\theta)_{ii}}\sum_{j=1}^{n}(\mathbf{Q}_\theta)_{ij}\eta_t^{(j)}, \tag{4.18}$$

where $(\mathbf{Q}_\theta)_{ij}$ is the i-th row and j-th column element of $\kappa^{-1}\mathbf{Q}_\theta$. Here, $\eta_t^{(-i)}$ is the collection of η_t values everywhere except $\mathring{s}^{(i)}$. The hyperparameter vector is defined as $\theta = (\kappa, \alpha)^T \in \mathbb{R}^2_{>0}$, where $\alpha = a - 4$. Fig. 4.2 shows the value of $\left(\kappa^{-1}\mathbf{Q}_\theta\right)_{ij}$ for one point along with its neighbors, graphically. The value of $(\mathbf{Q}_\theta)_{ij}$ is $(4+a^2)\kappa^{-1}$ as denoted at the center node of the graph in Fig. 4.2. That of $(\mathbf{Q}_\theta)_{ij}$ is $-2a\kappa^{-1}$ if j is one of the four closest neighbors of i in the vector 1-norm sense as illustrated by the graph in Fig. 4.2. Thus, the value of $(\mathbf{Q}_\theta)_{ij}$ is zero if j is not one of the twelve closest neighbors of i (or twelve neighbors whose 1-norm distance to the i-th location is less than or equal to 2). The equation in (4.18) states that the conditional expectation of $\eta_t^{(i)}$ given the value of η_t everywhere else (i.e., $\eta_t^{(-i)}$) can be determined just by knowing the value of η_t on the twelve closest neighbors (see more details in [63]). The resulting GMRF accurately represents a Gaussian random field with the Matérn covariance function as shown in [63]

$$G(r) = {\sigma_f}^2 \frac{2^{1-\rho}}{\Gamma(\rho)}\left(\frac{\sqrt{2\rho}r}{\ell}\right)^{\rho} K_\rho\left(\frac{\sqrt{2\rho}r}{\ell}\right), \tag{4.19}$$

where $K_\rho(\cdot)$ is a modified Bessel function [84], with order $\rho = 1$, a bandwidth $\ell = 1/h\sqrt{\frac{\alpha}{2}}$, and vertical scale ${\sigma_f}^2 = 1/4\pi\alpha\kappa$.

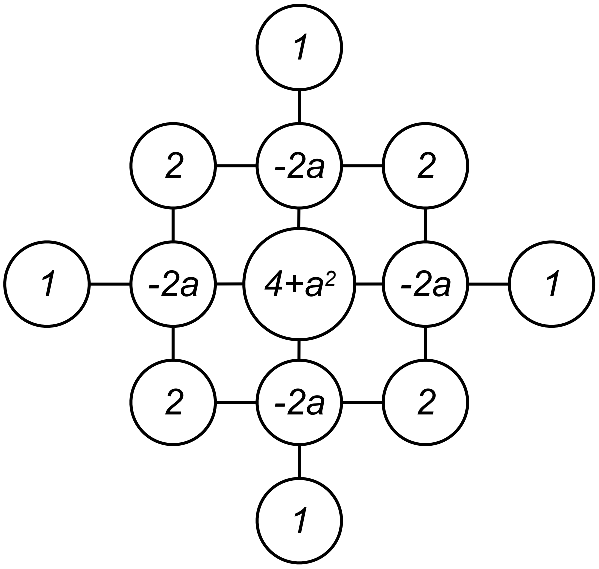


Figure 4.2: Elements of the precision matrix $\mathbf{Q}_\theta$ related to a given location.

The hyperparameter $\alpha > 0$ guarantees the positive definiteness of the precision matrix $\mathbf{Q}_\theta$. In the case where $\alpha = 0$, the resulting GMRF is a second-order polynomial intrinsic GMRF [88, 89].

From the presented model in (4.13), (4.15), and (4.16), the distribution of $\mathbf{z}_t$ given β_t and θ is

$$\mathbf{z}_t|\beta_t, \theta \sim \mathcal{N}\left(F_s\beta_t, \mathbf{Q}_\theta^{-1}\right), \tag{4.20}$$

where $F_s := (f(\mathring{s}^{(1)}), \cdots, f(\mathring{s}^{(n)}))^T \in \mathbb{R}^{n\times p}$.

In other words, $\mathbf{z}_t|\beta_t, \theta \sim \mathcal{GP}(F_s\beta_t, \mathbf{K}_\theta) \in \mathbb{R}^n$ is a non-zero mean Gaussian process. Here, the covariance matrix $\mathbf{K}_\theta$ is defined as inverse of the precision matrix (i.e., $\mathbf{K}_\theta = \mathbf{Q}_\theta^{-1}$). Note that the precision matrix is a positive definite matrix and invertible, and $(\mathbf{K}_\theta)_{ij} = \mathbb{C}\text{ov}(z_t^{(i)}, z_t^{(j)})$, where $(\mathbf{K}_\theta)_{ij}$ is the i, j-th element of the covariance matrix.

For simplicity, let us define $\mathcal{B}_t = \{\beta_t, \mathbf{q}_t, \mathbf{y}_t, \theta\}$. Using Gaussian process regression, the posterior distribution for $\mathbf{z}_t|\mathcal{B}_t \in \mathbb{R}^n$ is given by

$$\begin{aligned}\mu_{\mathbf{z}_t|\mathcal{B}_t} =& F_s\beta_t + \mathbf{K}_\theta H_t \left(H_t^T \mathbf{K}_\theta H_t + {\sigma_\epsilon}^2 I\right)^{-1} \left(\mathbf{y}_t - H_t^T F_s \beta_t\right), \\ \Sigma_{\mathbf{z}_t|\mathcal{B}_t} =& \mathbf{K}_\theta - \mathbf{K}_\theta H_t \left(H_t^T \mathbf{K}_\theta H_t + {\sigma_\epsilon}^2 I\right)^{-1} H_t^T \mathbf{K}_\theta.\end{aligned} \tag{4.21}$$

The basic idea behind the model introduced in (4.13), (4.15), and (4.16) stems from the space-time Kalman filter model proposed in [16]. The advantage of this spatio-temporal model with known hyperparameters is to make inferences in a recursive manner as the number of observations increases. The zero-mean Gaussian process represents a spatial structure by assuming that the difference between the parametric mean function and the dynamical environmental process is governed by a relatively large time scale. This formulation in turn makes the optimal prediction recursive in time.

In this chapter, however, uncertainties in the precision matrix and sampling positions are considered in a fully Bayesian manner. In addition, in contrast to [12, 16, 53], the GMRF with a sparse precision matrix is used to increase the computational efficiency.

4.5 Bayesian Predictive Inference

4.5.1 Uncertain hyperparameters and exact localization

In this section, we consider the problem of predicting a spatio-temporal random field, using successive noisy measurements sampled by a mobile sensor network. For a known covariance function, the prediction can be shown to be recursive [12] based on Gaussian process regression. The uncertainty in θ in a GMRF has been considered and its sequential prediction algorithms are derived in [116]. However, only the static field has been considered, i.e.

$\mu_t = \mu_0$. In this section, we use a Bayesian approach to make predictive inferences of the spatio-temporal random field $\mathbf{z}_t \in \mathbb{R}^n$ for the case with uncertain hyperparameters and the exact localization. To this end, we use the following assumptions A.1-A.5.

A.1 The spatio-temporal random field is generated by (4.13), (4.15), and (4.16).

A.2 The precision matrix $\mathbf{Q}_\theta$ is a given function of an uncertain hyperparameter vector θ.

A.3 The noisy measurements $\{\mathbf{y}_t\}$, as in (4.11), are continuously collected by robotic sensors in time t.

A.4 The sample positions $\{\mathbf{q}_t\}$ are measured precisely by robotic sensors in time t.

A.5 The prior distribution of the hyperparameter vector θ is discrete with a support $\Theta = \left\{\theta^{(1)}, \cdots, \theta^{(L)}\right\}$.

Remark 4.5.1. *A.1 and A.2 stem from the discretization of a Gaussian process as we described in Section 4.2. From the model in A.1, the zero-mean GMRF represents a spatial structure by assuming that the difference between the parametric mean function and the dynamical environmental process is governed by a relatively large time scale. However, this assumption could be rather strong for cases where data can be fitted better with temporally correlated GMRFs. A.3 is a standard assumption over the measured observations [53]. A.4 is rather strong but it will be relaxed to A.6 in Section 4.5.2 in order to deal with localization uncertainty. A.5 is from the discretization of the hyperparameter vector to replace an integration with a summation over possible hyperparameters.*

For notational simplicity, we denote the full latent field of dimension $n+p$ by $\mathbf{x}_t = (\mathbf{z}_t^T, \beta_t^T)^T$. Let's define $\mathcal{D}_{k:r} := \{\mathcal{P}_{k-1}, \mathbf{q}_{k:r}, \mathbf{y}_{k:r}\}$, where

$$\mathcal{P}_k = \left\{ \mu_{\mathbf{x}_k|\mathcal{D}_{1:k}}, \Sigma_{\mathbf{x}_k|\mathcal{D}_{1:k}} \} \cup \{ \mathbb{P}(\theta|\mathcal{D}_{1:k}) | \theta \in \Theta \right\},$$

and $\mathcal{P}_0$ is assumed to be known.

We formulate the first problem as follows.

Problem 4.5.2. *Consider the assumptions A.1-A.5. Our problem is to find the predictive distribution, mean, and variance of $\boldsymbol{x}_t$ conditional on $\mathcal{D}_{1:t}$.*

In what follows, we summarize the intermediate steps to obtain the solution to Problem 4.5.2.

Lemma 4.5.3. *Under assumptions A.1 and A.2, the predictive distribution of $\boldsymbol{x}_t$ conditional on the hyperparameter vector θ and the measurements $\mathcal{D}_{1:t-1}$ is Gaussian with the following mean and precision matrix*

$$\begin{aligned} \mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} &= \begin{pmatrix} F_s \mu_{\beta_t|\theta,\mathcal{D}_{1:t-1}} \\ \mu_{\beta_t|\theta,\mathcal{D}_{1:t-1}} \end{pmatrix}, \\ \boldsymbol{Q}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} &= \begin{pmatrix} \boldsymbol{Q}_\theta & -\boldsymbol{Q}_\theta F_s \\ -F_s^T \boldsymbol{Q}_\theta & F_s^T \boldsymbol{Q}_\theta F_s + \Sigma^{-1}_{\beta_t|\theta,\mathcal{D}_{1:t-1}} \end{pmatrix}, \end{aligned} \tag{4.22}$$

where $\mu_{\beta_t|\theta,\mathcal{D}_{1:t-1}} = A_t \mu_{\beta_{t-1}|\theta,\mathcal{D}_{1:t-1}}$ denotes the expectation of β_t conditional on θ and $\mathcal{D}_{1:t-1}$ and $\Sigma_{\beta_t|\theta,\mathcal{D}_{1:t-1}} = A_t \Sigma_{\beta_{t-1}|\theta,\mathcal{D}_{1:t-1}} A_t^T + B_t W B_t^T$ denotes the associated estimation error covariance matrix. For a given hyperparameter vector θ, (4.22) *provides the optimal prediction of the spatio-temporal field in time t using data up to time $t-1$.*

The following lemma is used to compute the posterior distribution of θ, recursively.

Lemma 4.5.4. *Under assumptions A.3 and A.4, the posterior distribution of the hyperparameter vector θ can be obtained recursively via*

$$\mathbb{P}(\theta|\mathcal{D}_{1:t}) \propto \mathbb{P}(\boldsymbol{y}_t|\theta, \mathcal{D}_{1:t-1}, \boldsymbol{q}_t)\, \mathbb{P}(\theta|\mathcal{D}_{1:t-1}), \tag{4.23}$$

where the distribution of $\boldsymbol{y}_t$ given $\{\theta, \mathcal{D}_{1:t-1}, \boldsymbol{q}_t\}$ is Gaussian with the following mean and variance

$$\begin{aligned} \mu_{\boldsymbol{y}_t|\theta,\mathcal{D}_{1:t-1},\boldsymbol{q}_t} &= M_t^T \mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}}, \\ \Sigma_{\boldsymbol{y}_t|\theta,\mathcal{D}_{1:t-1},\boldsymbol{q}_t} &= M_t^T \Sigma_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} M_t + {\sigma_\epsilon}^2 I, \end{aligned} \tag{4.24}$$

here $M_t^T = [H_t^T \;\; 0] \in \mathbb{R}^{N\times(n+p)}$.

Proof. The posterior distribution of θ given in (4.23) is computed by applying Bayes' rule on $\mathbb{P}(\theta|\mathbf{y}_t, \mathcal{D}_{1:t-1})$. The predictive statistics of $\mathbf{y}_t|\theta, \mathcal{D}_{1:t-1}$ are straightforward results of using (4.11). Note that $\mathbb{P}(\theta|\mathcal{D}_{1:t-1})$ is equal to $\mathbb{P}(\theta|\mathcal{D}_{1:t-1}, \mathbf{q}_t)$. □

Lemma 4.5.5. *Under assumptions A.1-A.4, the full conditional distribution of $\boldsymbol{x}_t$ for a given hyperparameter vector and data up to time t is*

$$\boldsymbol{x}_t|\theta, \mathcal{D}_{1:t} \sim \mathcal{N}(\mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}}, \boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}}),$$

where

$$\begin{aligned} \boldsymbol{Q}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} =& \boldsymbol{Q}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} + {\sigma_\epsilon}^{-2} M_t M_t^T, \\ \mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} =& \mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} + {\sigma_\epsilon}^{-2} \boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} M_t(\boldsymbol{y}_t - M_t^T \mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}}). \end{aligned} \tag{4.25}$$

Remark 4.5.6. *In order to keep computing the prediction error covariance matrix* $\boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}}$ *alone, the Woodbury lemma could be used to reduce the computational load as follows.*

$$\begin{aligned}\boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} =& \boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} - \boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} M_t \\ & \times \left(\sigma_\epsilon{}^2 I + M_t^T \boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} M_t\right)^{-1} M_t^T \boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}},\end{aligned} \tag{4.26}$$

where $\boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}}$ *can be computed with blockwise inversion using* (4.22)*,*

$$\boldsymbol{Q}^{-1}_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t-1}} = \begin{pmatrix} \boldsymbol{Q}_\theta^{-1} + F_s \Sigma_{\beta_t|\theta,\mathcal{D}_{1:t-1}} F_s^T & F_s \Sigma_{\beta_t|\theta,\mathcal{D}_{1:t-1}} \\ \Sigma^T_{\beta_t|\theta,\mathcal{D}_{1:t-1}} F_s^T & \Sigma_{\beta_t|\theta,\mathcal{D}_{1:t-1}} \end{pmatrix}.$$

The blockwise inversion needs to be updated only with $\Sigma_{\beta_t|\theta,\mathcal{D}_{1:t-1}}$.

The following theorem explicitly illustrates how the results of Lemma 4.5.4 and Lemma 4.5.5 lead to the predictive statistics of $\mathbf{x}_t$ under assumptions A.1-A.5, which will be the solution to Problem 4.5.2.

Theorem 4.5.7. *Under assumption A.5, the predictive distribution of* $\boldsymbol{x}_t|\mathcal{D}_{1:t}$ *is given by*

$$\mathbb{P}(\boldsymbol{x}_t|\mathcal{D}_{1:t}) = \sum_{\theta\in\Theta} \mathbb{P}(\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t})\,\mathbb{P}(\theta|\mathcal{D}_{1:t}), \tag{4.27}$$

where $\mathbb{P}(\theta|\mathcal{D}_{1:t})$ *and* $\mathbb{P}(\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t})$ *are given by Lemmas 4.5.4 and 4.5.5, respectively. The* *predictive mean and variance follow as*

$$\begin{aligned}
\mu_{\boldsymbol{x}_t|\mathcal{D}_{1:t}} &= \sum_{\theta\in\Theta} \mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}}\, \mathbb{P}(\theta|\mathcal{D}_{1:t}),\\
\Sigma_{\boldsymbol{x}_t|\mathcal{D}_{1:t}} &= \sum_{\theta\in\Theta} \left[\Sigma_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} + (\mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} - \mu_{\boldsymbol{x}_t|\mathcal{D}_{1:t}})(\mu_{\boldsymbol{x}_t|\theta,\mathcal{D}_{1:t}} - \mu_{\boldsymbol{x}_t|\mathcal{D}_{1:t}})^T\right] \mathbb{P}(\theta|\mathcal{D}_{1:t}).
\end{aligned} \tag{4.28}$$

Proof. The predictive mean and variance is obtained by marginalizing over the conditional distribution of θ given $\mathcal{D}_{1:t}$. The marginal mean and variance are $\mathbb{E}(Y) = \mathbb{E}(\mathbb{E}(Y|X))$ and $\mathbb{V}\text{ar}(Y) = \mathbb{E}(\mathbb{V}\text{ar}(Y|X)) + \mathbb{V}\text{ar}(\mathbb{E}(Y|X))$. Having $Y := \mathbf{x}_t|\mathcal{D}_{1:t}$ and $X := \theta|\mathcal{D}_{1:t}$ completes the proof of Theorem 4.5.7. □

Remark 4.5.8. *The optimal prediction of the spatio-temporal field $\boldsymbol{x}_t|\theta, \mathcal{D}_{1:t-1}$ using predictive statistics of $\boldsymbol{x}_{t-1}|\theta, \mathcal{D}_{1:t-1}$ is provided by Lemma 4.5.3. Lemma 4.5.5 provides the optimal estimator of $\boldsymbol{x}_t|\theta, \mathcal{D}_{1:t}$, using predictive statistics of $\boldsymbol{x}_t|\theta, \mathcal{D}_{1:t-1}$ which is given by Lemma 4.5.3. Using Lemma 4.5.3 and Lemma 4.5.5 sequentially we can update predictive statistics of $\boldsymbol{x}_t|\theta, \mathcal{D}_{1:t}$ for known hyperparameters. Lemma 4.5.4 gives us the posterior distribution of θ based on the measured data. Finally Theorem 4.5.7 provides the optimal Bayesian prediction of the spatio-temporal random field with a time varying mean function and uncertain hyperparameters by marginalizing θ over the conditional distribution of θ given $\mathcal{D}_{1:t}$.*

Remark 4.5.9. *Under assumption A.5, i.e., a discrete prior on the hyperparameter vector θ with a support $\Theta = \left\{\theta^{(1)}, \cdots, \theta^{(L)}\right\}$, the predictive distribution of $\boldsymbol{x}_t = (\boldsymbol{z}_t^T, \beta_t^T)^T$ (or the predictive mean and variance) conditional on $\mathcal{D}_{1:t}$ can be computed explicitly in a constant time as time t (or as the number of observations) increases in a recursive way. For a*

Algorithm 3 Sequential Bayesian predictive inference

```
Initialization:
1: initialize $F_s$
2: for $\theta \in \Theta$, initialize $\mathbf{Q}_\theta$, and compute $\mathbf{Q}_\theta^{-1}$
At time $t \in \mathbb{Z}_{>0}$, do:
1: obtain new observations $\mathbf{y}_t$ collected at current locations $\mathbf{q}_t$
2: find the map $M_t$ from $\mathbf{q}_t$ to spacial sites $\mathcal{Q}$, and compute radial basis values $F_{\mathbf{q}_t}$ in $\mathbf{q}_t$.
3: for $\theta \in \Theta$ do
4:     predict $\mu_{\mathbf{x}_t|\theta,\mathcal{D}_{1:t-1}}$ and $\mathbf{Q}_{\mathbf{x}_t|\theta,\mathcal{D}_{1:t-1}}$ using measurements up to time $t-1$, given by (4.22).
5:     compute $\mu_{\mathbf{x}_t|\theta,\mathcal{D}_{1:t}}$ and $\mathbf{Q}_{\mathbf{x}_t|\theta,\mathcal{D}_{1:t}}$ given by (4.25).
6:     compute $\mu_{\mathbf{y}_t|\theta,\mathcal{D}_{1:t-1},\mathbf{q}_t}$ and $\Sigma_{\mathbf{y}_t|\theta,\mathcal{D}_{1:t-1},\mathbf{q}_t}$ given by (4.24).
7:     calculate $\mathbb{P}(\theta|\mathcal{D}_{1:t})$ given by (4.23).
8: end for
9: compute the predictive mean and variance using (4.28).
```

continuous prior distribution $\mathbb{P}(\theta)$*, the summations in* (4.27) *and* (4.28) *need to be replaced by integrations with respect to* θ*, which need to be approximated. The proposed solution to the formulated problem is summarized by Algorithm 3.*

4.5.2 Uncertain hyperparameters and localization

In the previous section, we assumed that the localization data $\mathbf{q}_{1:t}$ is exactly known. However, in practice positions of sensor networks cannot be measured without noise. Instead, for example, there could be several probable possibilities inferred from the measured position. In this section, the proposed method in the previous section will be extended for the uncertain localization data. To the best of our knowledge, this is the first attempt to obtain predictive statistics for a case where both uncertainties in hyperparameters and localization are simultaneously included in a fully Bayesian perspective. In what follows, we present our idea in a problem statement and the corresponding solution.

In order to take into account the uncertainty in the sampling positions, we replace assumption A.4 with the following assumption A.6.

A.6 The prior distribution $\mathbb{P}(\mathbf{q}_t = \mathring{\mathbf{q}}_t^{(i)})$ is discrete with a support $\mathring{\mathbf{q}}_t^{(i)} \in \Omega(t)$, and

$$\Omega(t) = \left\{(\mathring{s}^{(k_1)}, \cdots, \mathring{s}^{(k_N)}) | (k_1, \cdots, k_N) \in \mathcal{L}(t)\right\}$$

, which is given at time t along with the corresponding measurement $\mathbf{y}_t$. Here, $\gamma(t) = |\mathcal{L}(t)|$ denotes the number of the probable possibilities for $\mathbf{q}_t$.

A straightforward consequence of Assumption A.6 is that the prior distribution $\mathbb{P}(\mathbf{q}_{k:r})$ is discrete with a support $\Omega(k:r) := \prod_{g=k}^{r} \Omega(g)$. In addition, $\mathcal{L}(k:r) := \prod_{g=k}^{r} \mathcal{L}(g)$ denotes the index in the support $\Omega(k:r)$, and $\gamma(k:r) := \prod_{g=k}^{r} \gamma(g)$ is the number of the probable possibilities for $\mathbf{q}_{k:r}$. Now we state the problem as follows.

Problem 4.5.10. *Consider assumptions A.1, A.2, A.3, A.5, and A.6. Our problem is to find the predictive distribution, mean and variance of $\boldsymbol{x}_t$ conditional on the prior $\mathcal{P}_0$ and the measurements $\boldsymbol{y}_{1:t}$.*

For the sake of conciseness, let us define $\mathcal{R}_{r:k} := \{\mathcal{P}_{r-1}, \mathbf{y}_{r:k}\}$. We then have that $\mathcal{R}_{r:k} \subset \mathcal{D}_{r:k}$, where we recall that $\mathcal{D}_{r:k} := \{\mathcal{P}_{r-1}, \tilde{\mathbf{q}}_{r:k}, \mathbf{y}_{r:k}\}$.

To solve Problem 4.5.10, we first seek for a way to compute the posterior distribution of $\mathbf{q}_t$ as summarized in the following lemma.

Lemma 4.5.11. *Consider $\mathbb{P}(\boldsymbol{y}_t|\theta, \mathcal{D}_{1:t-1}^{(n)}, \mathring{\boldsymbol{q}}_t^{(k)})$ given by (4.24) and $\mathbb{P}(\theta|\mathcal{D}_{1:t-1}^{(n)})$ given by (4.23), where $n \in \mathcal{L}(1:t-1)$, $k \in \mathcal{L}(t)$, and $\mathcal{D}_{1:t-1}^{(n)} := \left\{\mathcal{P}_0, \mathring{\boldsymbol{q}}_{1:t-1}^{(n)}, \boldsymbol{y}_{1:t-1}\right\}$. Under* *assumption A.5, we have*

$$\mathbb{P}(\boldsymbol{y}_t|\mathcal{D}^{(n)}_{1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_t) = \sum_{\theta\in\Theta} \mathbb{P}(\boldsymbol{y}_t|\theta, \mathcal{D}^{(n)}_{1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_t)\, \mathbb{P}(\theta|\mathcal{D}^{(n)}_{1:t-1}).$$

Under assumption A.6, the posterior distribution of $\boldsymbol{q}_{1:t} = \left(\mathring{\boldsymbol{q}}^{(n)}_{1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_t\right)$ *can be obtained, recursively, via*

$$\mathbb{P}\left(\mathring{\boldsymbol{q}}^{(n)}_{1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_t|\mathcal{R}_{1:t}\right) \propto \mathbb{P}(\mathring{\boldsymbol{q}}^{(n)}_{1:t-1}|\mathcal{R}_{1:t-1})\, \mathbb{P}(\boldsymbol{y}_t|\mathcal{D}^{(n)}_{1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_t)\, \mathbb{P}(\mathring{\boldsymbol{q}}^{(k)}_t). \tag{4.29}$$

Proof. The proof is a straightforward result of applying Bayes' rule. □

We now give the exact solution to Problem 4.5.10 as follows.

Theorem 4.5.12. *Consider the predictive distribution* $\mathbb{P}(\boldsymbol{x}_t|\mathcal{D}^{(i)}_{1:t})$ *given by Theorem 4.5.7 and the posterior* $\mathbb{P}\left(\mathring{\boldsymbol{q}}^{(i)}_{1:t}|\mathcal{R}_{1:t}\right)$ *given by Lemma 4.5.11, where* $\mathring{\boldsymbol{q}}^{(i)}_{1:t} \in \Omega(1:t)$ *and* $\mathcal{D}^{(i)}_{1:t} = \left\{\mathcal{P}_0, \mathring{\boldsymbol{q}}^{(i)}_{1:t}, \boldsymbol{y}_{1:t}\right\}$. *Under assumption A.6, the predictive distribution of* $\boldsymbol{x}_t|\mathcal{R}_{1:t}$ *can be obtained as follows.*

$$\mathbb{P}\left(\boldsymbol{x}_t|\mathcal{R}_{1:t}\right) = \sum_{i\in\mathcal{L}(1:t)} \mathbb{P}\left(\boldsymbol{x}_t|\mathcal{D}^{(i)}_{1:t}\right) \mathbb{P}\left(\mathring{\boldsymbol{q}}^{(i)}_{1:t}|\mathcal{R}_{1:t}\right). \tag{4.30}$$

Consequently, the predictive mean and variance are given by the formulas.

$$\begin{aligned}
\mu_{\boldsymbol{x}_t|\mathcal{R}_{1:t}} &= \sum_{i\in\mathcal{L}(1:t)} \mu_{\boldsymbol{x}_t|\mathcal{D}^{(i)}_{1:t}}\, \mathbb{P}\left(\mathring{\boldsymbol{q}}^{(i)}_{1:t}|\mathcal{R}_{1:t}\right), \\
\Sigma_{\boldsymbol{x}_t|\mathcal{R}_{1:t}} &= \sum_{i\in\mathcal{L}(1:t)} \left[\Sigma_{\boldsymbol{x}_t|\mathcal{D}^{(i)}_{1:t}} + \left(\mu_{\boldsymbol{x}_t|\mathcal{D}^{(i)}_{1:t}} - \mu_{\boldsymbol{x}_t|\mathcal{R}_{1:t}}\right)\right. \\
&\quad \left.\times \left(\mu_{\boldsymbol{x}_t|\mathcal{D}^{(i)}_{1:t}} - \mu_{\boldsymbol{x}_t|\mathcal{R}_{1:t}}\right)^T\right] \mathbb{P}\left(\mathring{\boldsymbol{q}}^{(i)}_{1:t}|\mathcal{R}_{1:t}\right).
\end{aligned} \tag{4.31}$$

Proof. The proof is similar to that of Theorem 4.5.7. Hence, the predictive mean and variance are obtained by marginalizing over the conditional distribution of $\mathbf{q}_{1:t}$. The marginal mean and variance are $\mathbb{E}(Y) = \mathbb{E}(\mathbb{E}(Y|X))$ and $\mathbb{V}\mathrm{ar}(Y) = \mathbb{E}(\mathbb{V}\mathrm{ar}(Y|X)) + \mathbb{V}\mathrm{ar}(\mathbb{E}(Y|X))$. Having $Y := \mathbf{x}_t|\mathcal{R}_{1:t}$ and $X := \mathbf{q}_{1:t}|\mathcal{R}_{1:t}$ proves Theorem 4.5.12. □

Remark 4.5.13. *The complexity of the proposed algorithm in Theorem 4.5.12 is proportional to the number of possibilities for $\boldsymbol{q}_{1:t}$. The result of Lemma 4.5.11 enables us to compute these probable possibilities recursively. However, the number of the non-zero probable combinations grows exponentially by a power of time. In other words, the number of possibilities for $\boldsymbol{q}_{1:t}$ is the multiplication of the numbers of possibilities for $\boldsymbol{q}_{1:t-1}$ and possibilities for $\boldsymbol{q}_t$. Note that this growing computational complexity of this exact solution, given in Theorem 4.5.12, to Problem 4.5.10 is prohibitive for sensor networks with limited resources.*

In what follows, we propose an approximation, with a controllable tradeoff between approximation error and complexity, to the exact solution given in Theorem 4.5.12 by including an option of ignoring uncertainties on past position data. This approximation will include a case of the exact solution with the maximal and original complexity. The idea is based on the fact that the estimation of $\mathbf{x}_t$ is more susceptible to the uncertainties in recently sampled positions as compared to old ones. To formulate our idea clearly, we present first the following results.

Lemma 4.5.14. *Using prior distribution of $\boldsymbol{x}_{t-m}$ and measured data $\boldsymbol{y}_{t-m+1:t}$, where $m \in \mathbb{Z}_{>0}$, the posterior distribution of $\boldsymbol{q}_{t-m+1:t}$ can be obtained recursively via*

$$\begin{aligned}&\mathbb{P}\left(\mathring{\boldsymbol{q}}^{(j)}_{t-m+1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_{t}|\mathcal{R}_{t-m+1:t}\right) \propto \\ &\quad \mathbb{P}(\mathring{\boldsymbol{q}}^{(j)}_{t-m+1:t-1}|\mathcal{R}_{t-m+1:t-1})\,\mathbb{P}(\boldsymbol{y}_t|\mathcal{D}^{(j)}_{t-m+1:t-1}, \mathring{\boldsymbol{q}}^{(k)}_{t})\,\mathbb{P}(\mathring{\boldsymbol{q}}^{(k)}_{t}).\end{aligned} \tag{4.32}$$

where $j \in \mathcal{L}(t-m+1:t-1)$, and $k \in \mathcal{L}(t)$.

Proof. The proof is a straightforward result of applying Bayes' rule. □

Theorem 4.5.15. *Consider $\mu_{\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}}$ and $\Sigma_{\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}}$ computed by Theorem 4.5.7. Under assumption A.6, the predictive statistics of $\boldsymbol{x}_t|\mathcal{R}_{t-m+1:t}$ are as follows.*

$$\begin{aligned}
\mu_{\boldsymbol{x}_t|\mathcal{R}_{t-m+1:t}} &= \sum_{h\in\mathcal{L}(t-m+1,t)} \mu_{\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}} \mathbb{P}\left(\mathring{\boldsymbol{q}}^{(h)}_{t-m+1:t}|\mathcal{R}_{t-m+1:t}\right),\\
\Sigma_{\boldsymbol{x}_t|\mathcal{R}_{t-m+1:t}} &= \sum_{h\in\mathcal{L}(t-m+1,t)} \left[\Sigma_{\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}} + \left(\mu_{\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}} - \mu_{\boldsymbol{x}_t|\mathcal{R}_{t-m+1:t}}\right)\right.\\
&\quad \left.\times\left(\mu_{\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}} - \mu_{\boldsymbol{x}_t|\mathcal{R}_{t-m+1:t}}\right)^T\right] \mathbb{P}\left(\mathring{\boldsymbol{q}}^{(h)}_{t-m+1:t}|\mathcal{R}_{t-m+1:t}\right).
\end{aligned} \tag{4.33}$$

Proof. The proof is similar to that of Theorem 4.5.12. □

To implement approximations to the predictive statistics of $\mathbf{x}_t|\mathcal{R}_{1:t}$ which are given by Theorem 4.5.12, we consider the following conditions.

C.1 For $1 \ll m \leq t$, we have that

$$\mathbb{P}(\mathbf{x}_t|\mathcal{R}_{1:t}) \approx \mathbb{P}(\mathbf{x}_t|\mathcal{R}_{t-m+1:t}) \tag{4.34}$$

C.2 For $1 \ll m \leq t$, $\mathcal{P}_t$ can be approximated by

$$\mathcal{P}_t \approx \left\{\mu_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}, \Sigma_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}\right\} \cup \left\{\mathbb{P}(\theta|\mathcal{R}_{t-m+1:t})|\theta\in\Theta\right\}. \tag{4.35}$$

Under conditions C.1 and C.2, it is natural for us to propose the following approximations.

$$\begin{aligned} \mu_{\mathbf{x}_t|\mathcal{R}_{1:t}} &\approx \mu_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}, \\ \Sigma_{\mathbf{x}_t|\mathcal{R}_{1:t}} &\approx \Sigma_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}. \end{aligned} \tag{4.36}$$

Remark 4.5.16. *In Theorem 4.5.15, the predictive statistics of $\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}$ are obtained from Algorithm 3 which is given in Section 4.5.1. The only difference is that we start from time $t-m+1$ instead of time 1 with $\mathcal{P}_{t-m}$ instead of $\mathcal{P}_0$. Note that without condition C.2 we cannot use Algorithm 3 to calculate the statistics of $\boldsymbol{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}$. The proposed approximation for the case of uncertain localization in* (4.36) *is quite different from a mere truncation of old data in the sense that past measurements still affect the current estimation through the approximately updated prior information using* (4.35)*. Note that we update the prior information from $\mathcal{P}_{t-m}$ to $\mathcal{P}_t$ with the cumulative data collected from time $t-m+1$ up to time t, which is different from only using truncated observations. The proposed approximation for the formulated problem is summarized by Algorithm 4.*

To further understand the nature of the proposed approximation, consider the following two extreme special cases.

Corollary 4.5.17. *As a special case of Theorem 4.5.15 for $m = 1$, the posterior distribution of $\boldsymbol{q}_t$ can be obtained via*

$$\mathbb{P}(\mathring{\boldsymbol{q}}^{(k)}_t|\mathcal{P}_{t-1}, \boldsymbol{y}_t) \propto \mathbb{P}(\boldsymbol{y}_t|\mathcal{P}_{t-1}, \mathring{\boldsymbol{q}}^{(k)}_t)\, \mathbb{P}(\mathring{\boldsymbol{q}}^{(k)}_t), \tag{4.37}$$

where $k \in \mathcal{L}(t)$. The predictive distribution of $\boldsymbol{x}_t|\mathcal{R}_{1:t}$ can be approximated in a constant *time as time t increases in a sequential way.*

$$\mathbb{P}\left(\boldsymbol{x}_t|\mathcal{R}_{1:t}\right) \approx \mathbb{P}\left(\boldsymbol{x}_t|\mathcal{P}_{t-1}, \boldsymbol{y}_t\right),$$

where

$$\mathbb{P}\left(\boldsymbol{x}_t|\mathcal{P}_{t-1}, \boldsymbol{y}_t\right) = \sum_{k \in \mathcal{L}(t)} \mathbb{P}\left(\boldsymbol{x}_t|\mathcal{P}_{t-1}, \mathring{\boldsymbol{q}}_t^{(k)}, \boldsymbol{y}_t\right) \mathbb{P}\left(\mathring{\boldsymbol{q}}_t^{(k)}|\mathcal{P}_{t-1}, \boldsymbol{y}_t\right) \tag{4.38}$$

and the posterior $\mathbb{P}\left(\mathring{\boldsymbol{q}}_t^{(k)}|\mathcal{P}_{t-1}, \boldsymbol{y}_t\right)$ is given by (4.37)*. Consequently, the predictive mean $\mu_{\boldsymbol{x}_t|\mathcal{R}_{1:t}}$ and variance $\Sigma_{\boldsymbol{x}_t|\mathcal{R}_{1:t}}$ can be approximated by $\mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\boldsymbol{y}_t}$ and $\Sigma_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\boldsymbol{y}_t}$, respectively, i.e.,*

$$\begin{aligned}
\mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\boldsymbol{y}_t} &= \sum_{k \in \mathcal{L}(t)} \mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\mathring{\boldsymbol{q}}_t^{(k)},\boldsymbol{y}_t} \mathbb{P}\left(\mathring{\boldsymbol{q}}_t^{(k)}|\mathcal{P}_{t-1}, \boldsymbol{y}_t\right), \\
\Sigma_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\boldsymbol{y}_t} &= \sum_{k \in \mathcal{L}(t)} \Bigg[\Sigma_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\mathring{\boldsymbol{q}}_t^{(k)},\boldsymbol{y}_t} + \left(\mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\mathring{\boldsymbol{q}}_t^{(k)},\boldsymbol{y}_t} - \mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\boldsymbol{y}_t}\right) \\
&\quad \times \left(\mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\mathring{\boldsymbol{q}}_t^{(k)},\boldsymbol{y}_t} - \mu_{\boldsymbol{x}_t|\mathcal{P}_{t-1},\boldsymbol{y}_t}\right)^T\Bigg] \mathbb{P}\left(\mathring{\boldsymbol{q}}_t^{(k)}|\mathcal{P}_{t-1}, \boldsymbol{y}_t\right).
\end{aligned} \tag{4.39}$$

Corollary 4.5.18. *For another special case with $m = t$, Theorem 4.5.15 becomes Theorem 4.5.12.*

Remark 4.5.19. *For a fixed $m \in \mathbb{Z}_{>0}$, Algorithm 4 is scalable as time t (or the number of observations) increases. In our approach, the level of the approximation can be controlled by users by selecting a tradeoff (or by choosing a value for m) between the approximation error and complexity. Most simplistic approximation can be obtained by choosing $m = 1$ as in Corollery 4.5.17. The original exact solution with maximal complexity is recovered by*

Algorithm 4 Sequential Bayesian predictive inference approximation with uncertain localization.

```
At time $t \in \mathbb{Z}_{>0}$, do:
1: obtain new observations $\mathbf{y}_t$ along with the probabilities for locations $\mathbb{P}(\mathbf{q}_t)$
2: for $\mathring{\mathbf{q}}_t^{(h)} \in \Omega(t-m+1:t)$ do
3:     predict $\mu_{\mathbf{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}}$, $\Sigma_{\mathbf{x}_t|\mathcal{D}^{(h)}_{t-m+1:t}}$ and $\mathbb{P}(\mathbf{y}_t|\mathcal{D}^{(j)}_{t-m+1:t-1}, \mathring{\mathbf{q}}_t^{(k)})$ using Algorithm 3
4:     compute $\mathbb{P}\left(\mathring{\mathbf{q}}^{(h)}_{t-m+1:t}|\mathcal{R}_{t-m+1:t}\right)$ using Lemma 4.5.14.
5: end for
6: compute $\mu_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}$ and $\Sigma_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}$, using Theorem 4.5.15.
7: use following approximation to update estimations
       $\mu_{\mathbf{x}_t|\mathcal{R}_{1:t}} \approx \mu_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}},$
       $\Sigma_{\mathbf{x}_t|\mathcal{R}_{1:t}} \approx \Sigma_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}},$
       $\mathcal{P}_t \approx \left\{\mu_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}, \Sigma_{\mathbf{x}_t|\mathcal{R}_{t-m+1:t}}, \mathbb{P}(\Theta|\mathcal{R}_{t-m+1:t})\right\}.$
```

selecting $m = t$ as shown in Corollery 4.5.18.

4.5.3 Complexity of Algorithms

In this section, we discuss complexity aspects of the proposed algorithms. For a fixed number of the radial basis functions (i.e., p) and a fixed number of the special sites (i.e., n), the computational complexity of Algorithm 3 is dominated by (4.25). The complexity of Algorithm 3 in each time step is $\mathcal{O}(LN^2)$, where L is the number of possible hyperparameter vectors and N is the number of agents. The complexity of Algorithm 4 in time t is $\mathcal{O}\left(\gamma(t-m+1:t)\right)$ times the complexity of Algorithm 3 for m time steps. Hence, the complexity of Algorithm 4 in time t is $\mathcal{O}\left(\gamma(t-m+1:t)LN^2M\right)$.

The numbers of special sites and radial basis functions affect the complexity of Algorithm 3 as well. The complexity of the three cases is $\mathcal{O}(n^3)$ with respect to the number of special sites due to the matrix inversion. Thus, if we use finer grids we increase the number of special sites, i.e. n, and the complexity increase cubical. For a fixed set of L, n and N, the complexity of Algorithm 3 with respect to p is $\mathcal{O}(p^3)$.

4.6 Simulation Results

In this section, we demonstrate the effectiveness of the proposed sequential Bayesian inference algorithms using a numerical experiment. Consider a scenario, where mobile sensing agents can only move on a regular lattice, denoted by $\mathcal{G}$, which is shown with circles in Fig. 4.3. This reachable set of robotic sensors is a subset of the spatial sites (i.e., $\mathcal{G} \subset \mathcal{Q}$). Note that $\mathcal{Q}$ is a finer grid graph on which the GMRF is realized. The spatial sites are shown as intersections between dashed lines in Fig. 4.3. Assume that some agents measure their true positions exactly while others only know a finite set of probable positions with the associated probabilities. Fig. 4.3 shows an illustrative configuration where agents 1 and 3 measure sampling positions exactly while agent 2 knows only a set of uncertain possibilities (as shown with $2a, 2b, 2c$, and $2d$) for a given measured sampling position (a solid star). The prior probability that the true sampling position of agent 2 is located in $2a$ is proportional to the rectangular area between the star and $2d$, i.e., $\frac{2}{16}$. The other prior probabilities of $2b$, $2c$, and $2d$ are defined in the same way, which provides a high probability for a node whose location is close to the observed sampling location. This way of assigning priors guarantees that the sum of prior probabilities is equal to one. In this simulation, we allow two agents get measurements at the same location and time. Repeatedly sampled positions are denoted by large circles in Fig. 4.5-(g).

In this illustrative example, we use the spatio-temporal field introduced in Section 4.4.

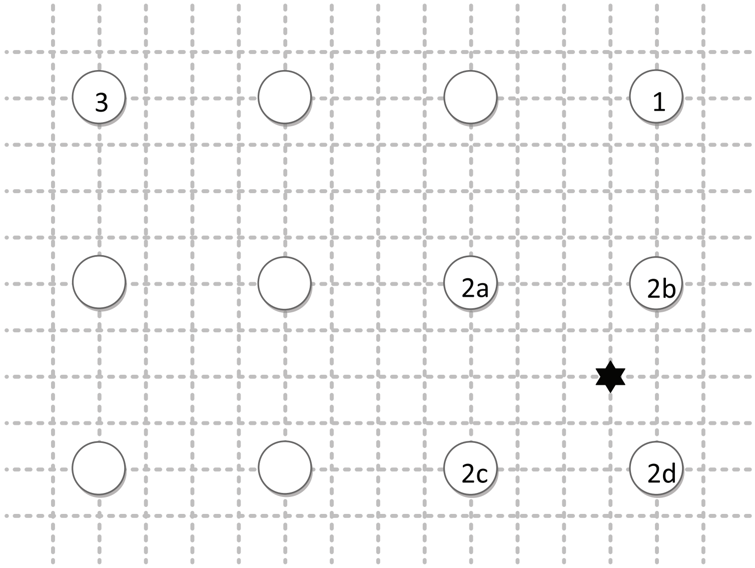


Figure 4.3: Agents 1 and 3 measure their exact sampling positions while agent 2 has a noisy sampling position measured in the star location. Four possible sampling positions for the true sampling position of agent 2 are $2a, 2b, 2c$ and $2d$ with the prior probabilities $\frac{2}{16}, \frac{6}{16}, \frac{2}{16}$, and $\frac{6}{16}$, respectively.

The spacial sites in $\mathcal{Q}$ consist of 51×51 grid points, i.e., $n = 2601$, uniformly distributed over the surveillance region $[-25, 25] \times [-25, 25]$. The time-varying mean function μ_t consists of ten radial basis functions (i.e., $p = 10$), which are defined as follow.

$$f_j(q^{[i]}) = \exp\left(-\frac{\|q^{[i]} - \xi_j\|^2}{2\sigma_j^2}\right), \quad j \in \{1, \cdots, p\}, \tag{4.40}$$

where σ_j is the bandwidth and ξ_j is the center location of the j-th radial basis function. The first radial basis function has an infinity bandwidth (i.e., $\sigma_1 = \infty$) to represent the average of the field, and the others have a bandwidth equal to $\sigma_j = 15$. The centers of radial basis functions are $\{(0, 0)\} \cup \{-15, 0, 15\} \times \{-15, 0, 15\}$. The prior distribution of β_0 is chosen to be $\beta_0 \sim \mathcal{N}(0, 5I)$. The time evolution of β_t is modeled by (4.15), where the state matrix A_t and the input matrix B_t are given by $0.95I_{10\times10}$ and $0.5I_{10\times10}$, respectively. The input disturbance variance is known to be $W = I_{10\times10}$.

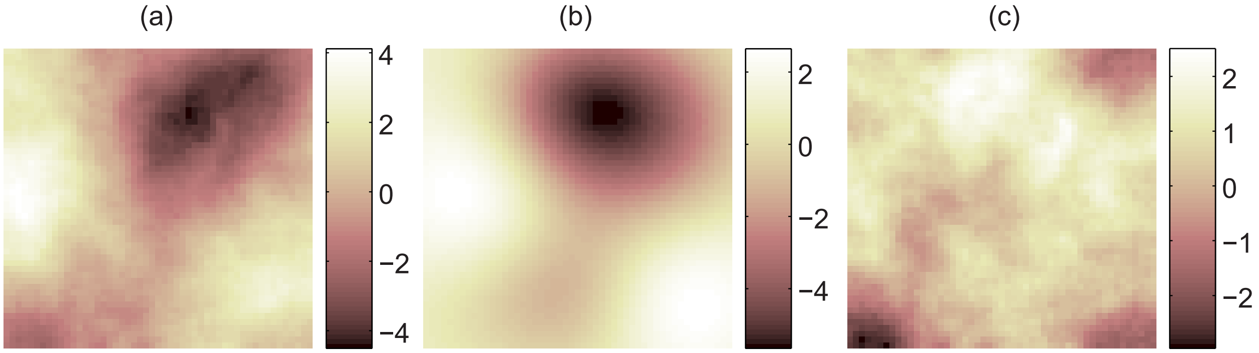


Figure 4.4: (a) The realized spatio-temporal field at time $t = 2$ is the summation of (b) the smooth mean field and (c) the GMRF as described in (4.13).

We choose the precision matrix $\mathbf{Q}_{x|\theta}$ with hyperparameters $\alpha = 0.01$ equivalent to a bandwidth $\ell = \sqrt{2}/\sqrt{\alpha} \approx 14.14$, and $\kappa = 10$ equivalent to ${\sigma_f}^2 = 1/4\pi\alpha\kappa \approx 0.8$. The measurement noise variance ${\sigma_\epsilon}^2 = 0.1$ is assumed to be known.

The numerically simulated field $\mathbf{z}_t$ from our model in (4.13) is shown in Fig. 4.4-(a). From the same realization, the mean function μ_t and the GMRF η_t, as in (4.13), are shown in Fig. 4.4-(b) and Fig. 4.4-(c), respectively.

Ten mobile sensing agents, i.e, $N = 10$, take measurements at time $t \in \{1, 2, \cdots, 10\}$, where only three agents are assumed to have uncertain sampling positions. In Figs. 4.5-(g), (h), and (i), true, noisy, and probable sampling positions are shown in circles, stars, and corners of squares, respectively.

The prediction results are summarized for three methods of prediction as follows.

- *Case 1:* Figs. 4.5-(a), (d), and (g) show the prediction, prediction error variance, and squared (empirical) error fields, using Algorithm 3 with exact sampling positions. With the true sampling positions, the best prediction quality is expected for this case.

- *Case 2:* Figs. 4.5-(b), (e), and (h) show the resulting fields, by applying Algorithm 3 naively to measured sampling positions including noisy locations. The results clearly illustrate that naively applying Algorithm 3 to noisy sampling positions can potentially distort prediction at a significant level as shown in Figs. 4.5-(b) and (h).

- *Case 3:* Figs. 4.5-(c), (f), and (i) show the resulting fields, by applying Algorithm 4 with $m = 1$. In particular, we use the result from Corollary 4.5.17 (i.e., $m = 1$) to evaluate the effectiveness of the proposed approximation in marginalizing the uncertainty in sampling positions. The resulting prediction quality is much improved as compared to Case 2 and is even comparable to the result for Case 1.

Cases 1, 2, and 3 will be also applied to realistic experimental data in Section 4.7.

Fig. 4.6-(a) shows the RMS errors of the estimated β_t for Cases 1, 2, and 3 in time. The RMS errors are defined as follows.

$$\mathrm{RMS}(t) = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(\mu_{\beta_t^{[i]}} - \beta_t^{[i]})^2}. \tag{4.41}$$

Fig. 4.6-(a) clearly illustrates that the RMS errors of Cases 1 and 3 are similarly lower than that of Case 2 for all time.

Now we examine the convergence rate of hyperparameters. The prior distribution of the hyperparameter vector θ is discrete with a support

$$\Theta = \{(\kappa, \alpha), (0.1\kappa, \alpha), (10\kappa, \alpha), (\kappa, 0.1\alpha), (\kappa, 10\alpha)\},$$

along with the corresponding uniform probabilities $\{0.2, 0.2, 0.2, 0.2, 0.2\}$. As shown in Fig. 4.6-(b), the updated posterior probabilities of the hyperparameters (κ, α) converge to the true hyperparameters for Case 1 (blue circles) and Case 3 (red squares), as the number of observations increases. However, Case 2 (green stars) does not converge to the true ones.

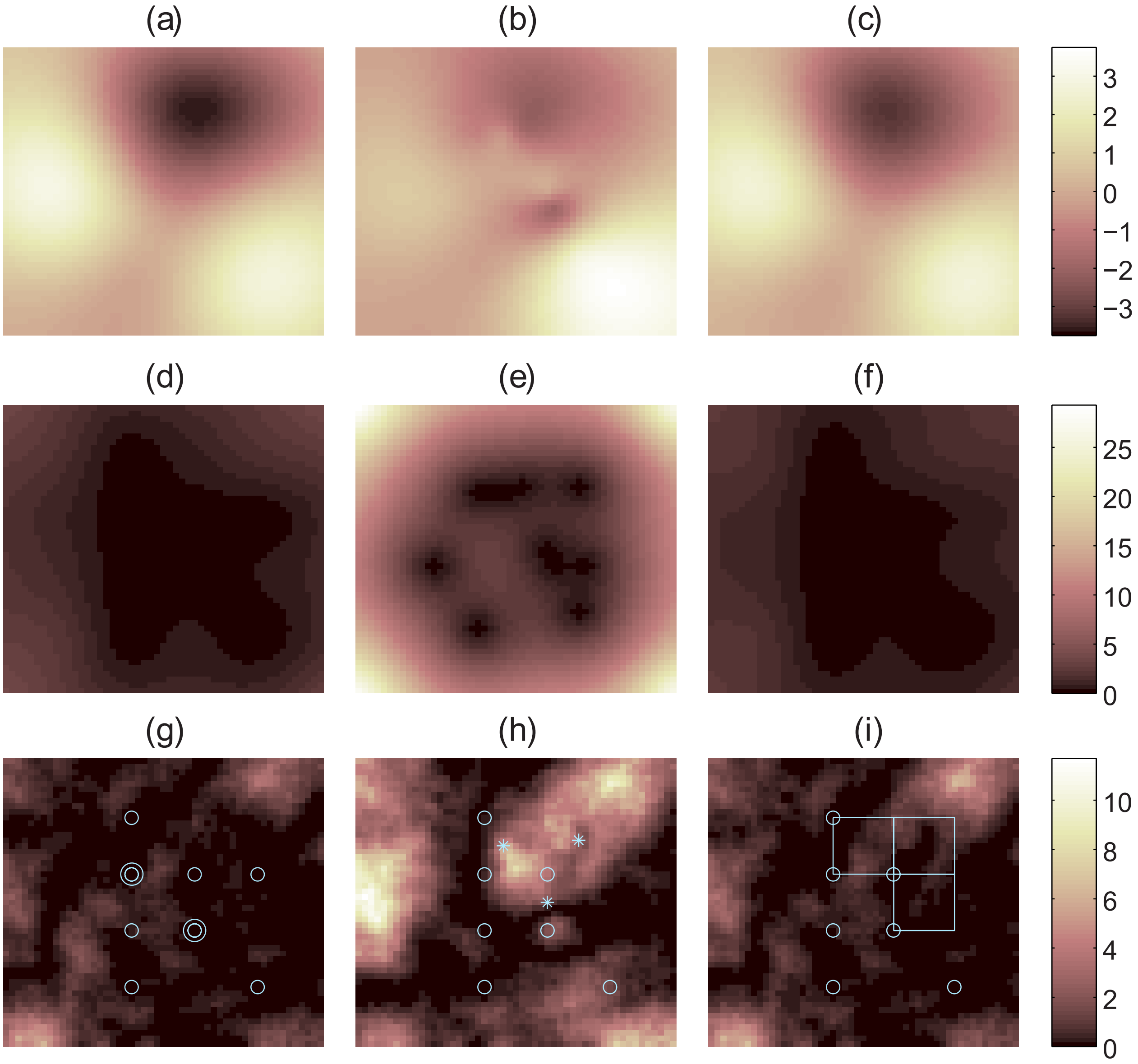


Figure 4.5: The prediction results of Cases 1, 2, and 3 at time $t = 2$ are shown in the first, second, third columns, respectively. The first, second, and third rows correspond to the prediction, prediction error variance, and squared empirical error fields between the prediction and true fields. True, noisy, and probable sampling positions are shown in circles, stars, and corners of squares, respectively.

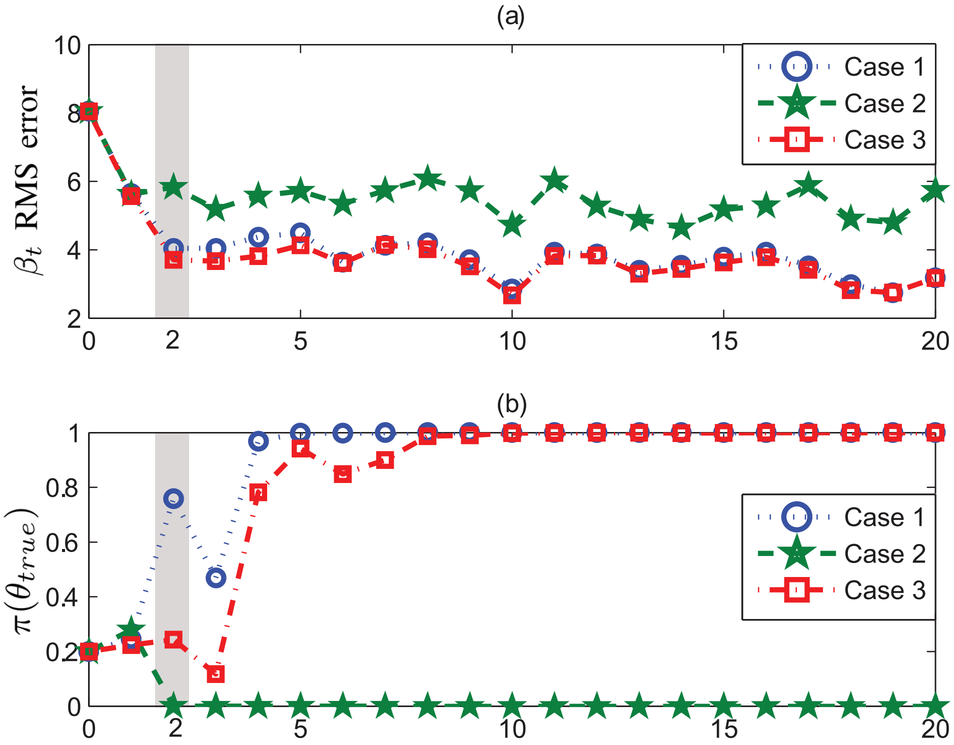


Figure 4.6: (a) The RMS estimation error of β_t v.s. time and (b) the posterior probability of the true hyperparameter vector v.s. time. The results for Cases 1, 2, and 3 are shown in blue circles, green stars, and red squares, respectively.

Note that the results in Figs. 4.4 and 4.5 are obtained at time $t = 2$. The corresponding values for $t = 2$ in Fig. 4.6 are highlighted by a gray area.

Fig. 4.7 shows the controllable tradeoff between approximation error and complexity. The complexity of Algorithm 4 increases exponentially with respect to m. The predicted field is compared between Case 1, which is the best prediction quality expected, and Case 3. Clearly, by increasing m, the mean square difference between Case 1 and Case 3 decreases. However, this improvement costs exponentially increasing computational load.

Fig. 4.8 shows the effect of increasing number of observations with uncertain sampling positions on the three cases. Here we assume that we have seven observations with known true sampling positions plus a few observations with uncertain sampling positions. Clearly the RMS error decreases in Case 1 and Case 3 by adding new observations with true and

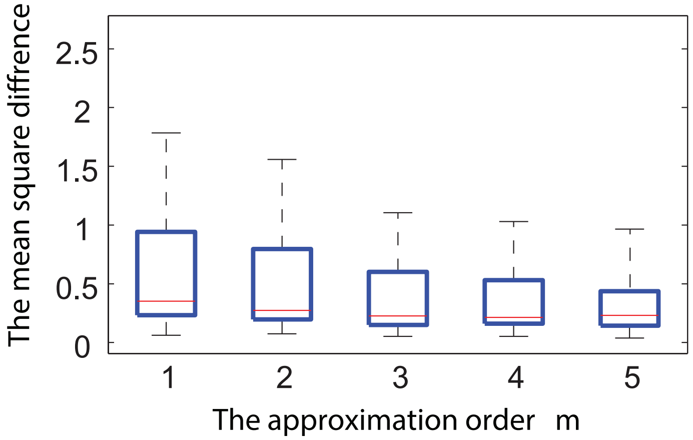


Figure 4.7: The mean square difference between Case 1 and Case 3 is shown for the different approximation orders $m = 1, \cdots, 5$. On each box, the central mark is the median, the upper and lower edges of the box are the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points.

uncertain sampling positions, respectively. On the other hand, as shown in Fig. 4.8, adding new observations with noisy sampling positions could increase RMS error for Case 2.

In this example, the fixed running time using Matlab R2009b (MathWorks) on a PC (2.4 GHz Intel Core 2 Duo Processor) is about 3 minutes for the proposed approximation ($m = 1$) which is fast enough for real world implementation. In addition, the quality of prediction by Algorithm 4 with a finite m could be as good as one using Algorithm 3 with the true sampling positions.

4.7 Experimental Results

Similarly to [82], a vision-based robotic sensor was built by the authors to validate the proof of concept for the proposed methods. The wireless mobile robot is equipped with two motorized wheels, a micro-controller, and a 360 degree omnidirectional camera, a motion sensor, a wireless receiver, and transmitter. The omnidirectional camera is homemade from a cheap Wi-Fi remote CCD camera (Ai-Ball Treck®) and a globe mirror. The vision images

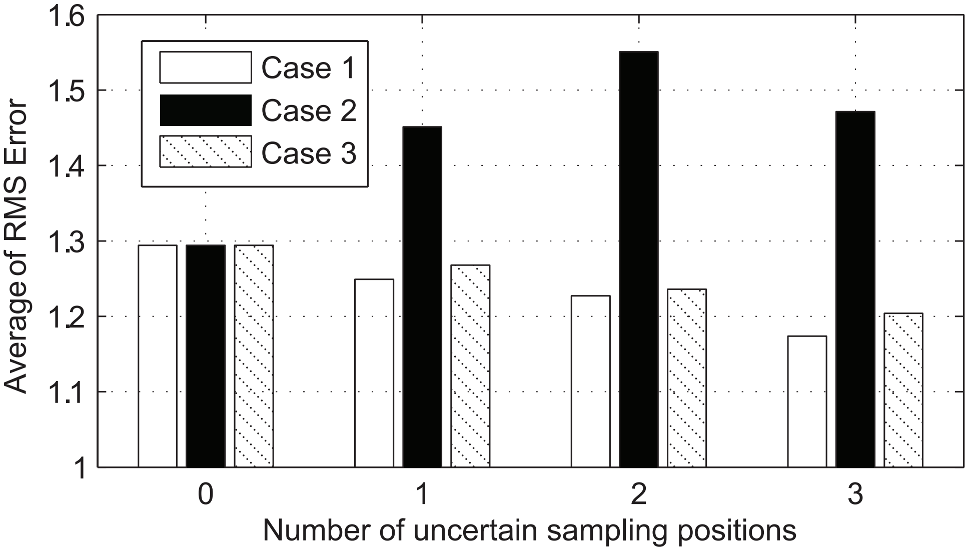


Figure 4.8: The number of observations with uncertain sampling position on the three different cases is shown on the horizontal axis. The field prediction RMS error for Case 1, Case 2 and Case 3 are shown with white, black, and hatched bars, respectively.

of the 360 degree environment around the robot produced by the omnidirectional camera are steamed via 802.11 b/g Wi-Fi interface to a laptop for image processing (see Fig. 4.9).

In this section, we apply the proposed prediction algorithms to real experimental data. Fig. 4.10 shows our experimental setup in which a redness intensity field is sampled by the captured images from the CCD camera on top of the mobile robot. The CCD camera captures 360 degree images. The redness intensity is computed by simply averaging the red component of the RGB picture. Noisy measurements are sampled at random sampling positions by our robot. The position of the robot has been measured by an image processing software which is built by the authors and the true sampling positions are obtained manually. The objective is to predict the redness intensity field using the scalar field model proposed in Section 4.4. Note that each image has a lot of information. However, we only used the redness out of each image as a scalar value of interest. In the future, we plan to extend this experiment for multivariate random fields for multiple features from each image. Here we are

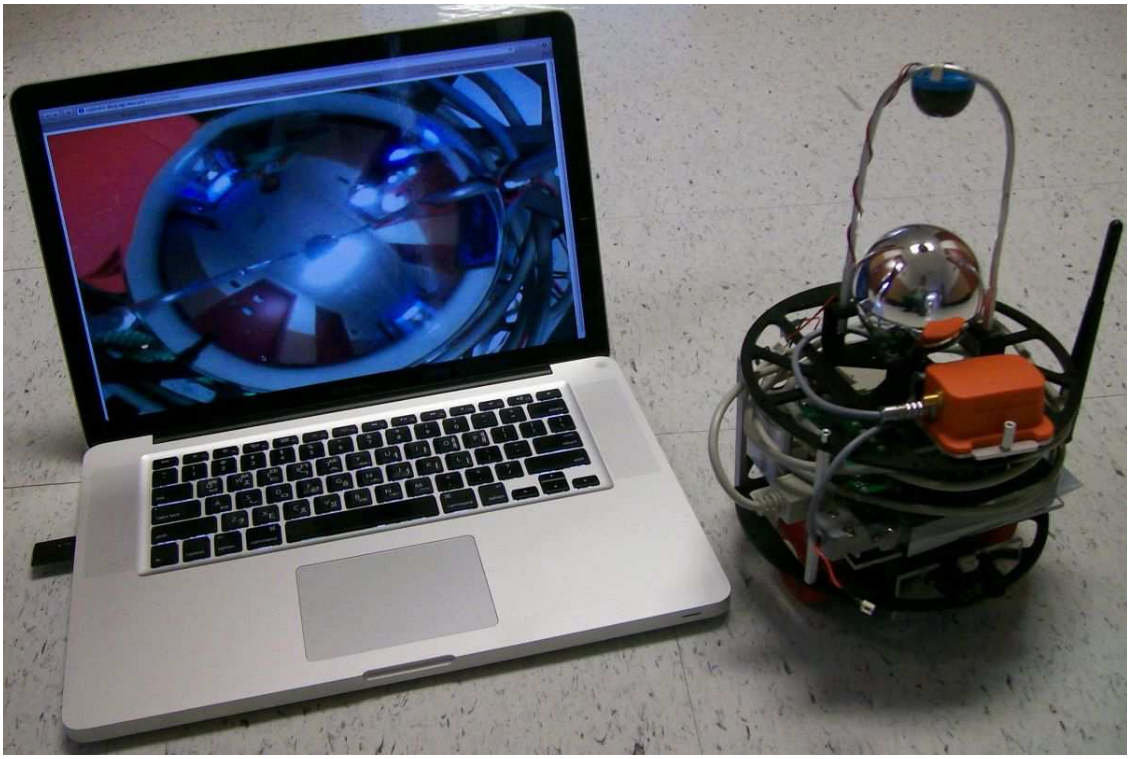


Figure 4.9: A vision-based robot is built by the authors, and equipped with a 360 degree omnidirectional camera.

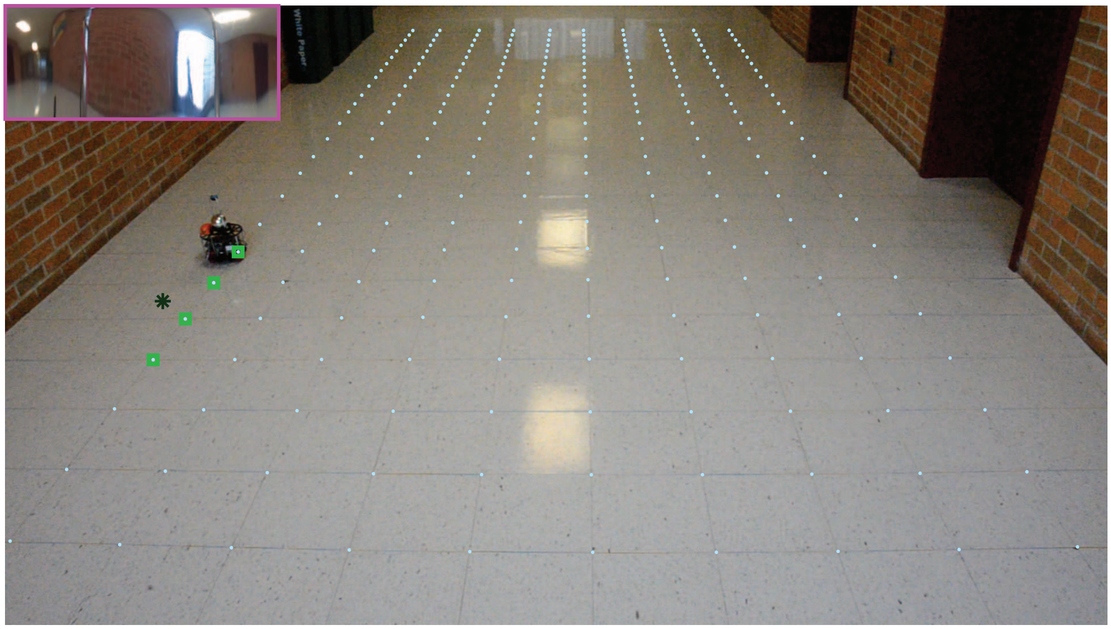


Figure 4.10: The experimental setup is shown. The measured position of the robot and 4 possible sampling positions are shown in dark green star and light green squares, respectively. The spacial sites are marked with aqua blue dots on the ground. The panoramic view of the robot is pictured on the upper left hand side of the figure.

considering two different scenarios. First, we consider a static field and then a time-varying field with a moving person in the surveillance region.

4.7.1 The Static Experiment

In this study, the spatial sites in $\mathcal{Q}$ are considered to be 10×26 grid points, i.e., $n = 260$. The grid points are shown in Fig. 4.10 with aqua-blue dots. The mean function μ_t consists of only one radial basis function that keeps moving average of the field. Note that this basis function can model the changes in the brightness of the images caused by the slow changes of the environment lights. The center of the radial basis function is $(5, 13)$, and its bandwidth σ_1 is ∞. The prior distribution of the hyperparameter vector θ is chosen to be discrete with a support $\Theta = \{25, 50, 100, 200, 400\} \times \{0.1, 0.2, 0.4, 0.8, 1.6\}$ and the associated uniform probabilities. The measurement noise variance $\sigma_\epsilon = 0.01$ is estimated.

To demonstrate the usefulness of our model in (4.13), (4.15), and (4.16), and our prediction algorithm, the mobile robot measures eighty samples, i.e., $N = 80$, where only nine sampling positions are uncertain with four possibilities each. Thus there is 4^9 possible combinations for this set of sampling positions. After two set of observations, the resulting posterior probabilities of the hyperparameters for Case 1 is shown in Fig. 4.11. Fig. 4.11 shows the estimated hyperparameters converge to $\kappa = 100$ and $\alpha = 0.8$, where is equivalent with $\ell = 1.58$ and $\sigma_f = 0.0315$.

Similarly to Section 4.6, the prediction and prediction error variance are computed for Cases 1, 2, and 3. However, since it is a realistic experiment, the ground truth of the field of interest is not available and so we are not able to compute the empirical error. The prediction and the prediction error variance using true sampling positions (Case 1), are shown in Figs. 4.12-(a) and (d), respectively. Figs. 4.12-(b) and (e) show the resulting fields, by applying

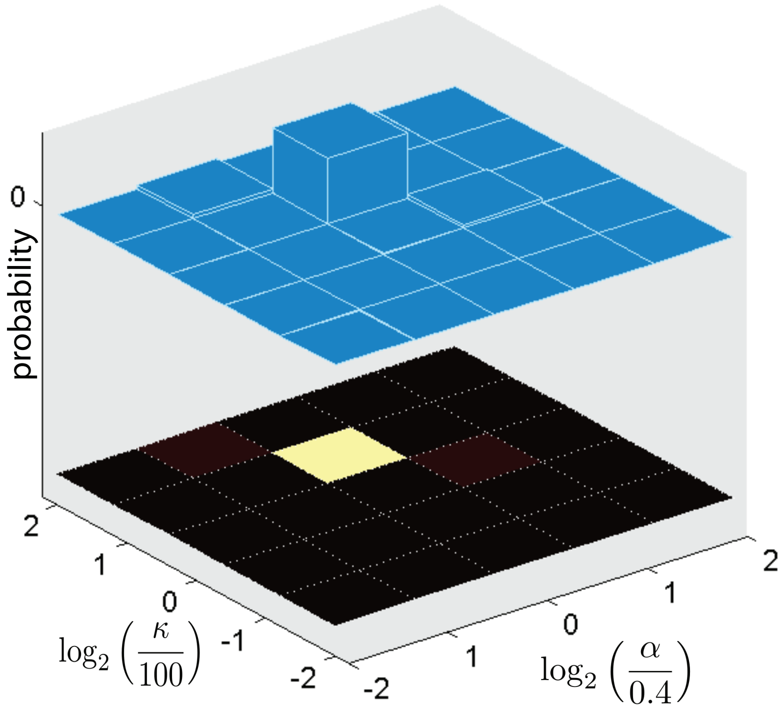


Figure 4.11: The posterior probability of the hyperparameter vector at $t = 2$.

Algorithm 3 naively to measured sampling positions including noisy locations (Case 2).

Figs. 4.12-(c) and (f) show the results by applying Algorithm 4 with $m = 1$ (Case 3). The results confirm that the quality of the prediction in Case 3 is not much compromised as compared to Case 1 and demonstrate the capability of our proposed algorithm to deal with uncertain sampling positions.

4.7.2 The Dynamic Experiment

In this scenario, the spatial sites are the same as in the previous experiment. The mean function μ_t consists of twenty nine radial basis functions. The centers of radial basis functions are $\{(5, 13)\} \cup \{1, 4, 7, 10\} \times \{1, 5, 9, 13, 17, 21, 25\}$. The time evolution of β_t is modeled by (4.15), where the state matrix A_t and the input matrix B_t are given by I and $0.05I$, respectively. The first radial basis function has an infinity bandwidth (i.e., $\sigma_1 = \infty$) to represent the average of the field, and the others have a bandwidth that is equal to $\sigma_j = 4$.

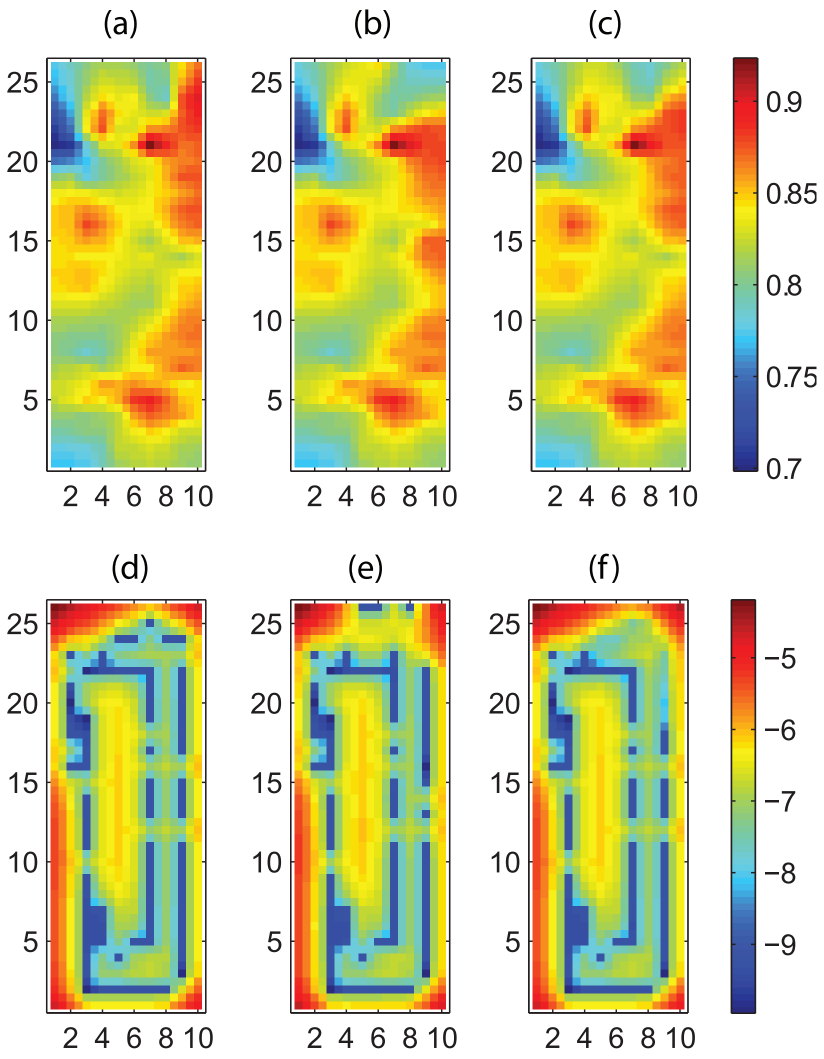


Figure 4.12: The prediction results of Cases 1, 2, and 3 at time $t = 2$ are shown in the first, second, third columns, respectively. The first and second rows correspond to the predictions and the natural logarithm of the prediction error variance.

In this experiment, a person moves in the surveillance region while the robot is collecting observations. The robot collects 57, 40, 31, and 45 observations corresponding to times $t = 1, 2, 3$, and 4, respectively. The first column in Fig. 4.13 shows the positions of moving robot and person in the domain. As mentioned, the position of the robot has been measured with a fixed camera and help of an image processing software. Sometimes, the robot is not visible by the positioning system since the moving person blocks the visual contact between the robot and the positioning system. The black areas in the second column of Fig. 4.13

show the blind spots of the positioning system. Note that other positioning systems like GPS also have blind spots such as GPS denied areas. Therefore, when the robot moves to a blind spot the position cannot be determined precisely. In this case, we assign different probabilities on multiple sampling positions.

True sampling positions in each time step are shown with blue dots in the second column of Fig. 4.13 while just blue dots in the white area are measured through the positioning system and the positions in the blind spots are recorded manually for a comparison purpose.

The prediction results of Case 1 and Case 3 are compared with a trivial method of prediction defined as follows.

- *Case 4:* The fifth column of Fig. 4.13 shows the resulting fields, by applying Algorithm 3 on just observed sampling positions. Here all the observations whose sampling positions are uncertain are discarded. In particular, $12, 3, 2,$ and 4 observations have been discarded for time $t = 1, 2, 3$ and 4, respectively.

The predicted field using true sampling positions (Case 1) and uncertain sampling positions (Case 3) are shown in the third and forth columns of Fig. 4.13, respectively. Since in the fully automated experiment true sampling positions in the blind spot are not available, we used the proposed algorithm in this chapter to deal with uncertain sampling positions in the blind spot areas. The predicted field simply by discarding uncertain sampling positions is shown in the fifth column of Fig. 4.13. Clearly, the results obtained for Case 3 with $m = 1$, is comparable to the result for Case 1 that is the best prediction quality expected among all cases. Thus, the experimental result demonstrates the effectiveness of the proposed algorithm in this chapter.

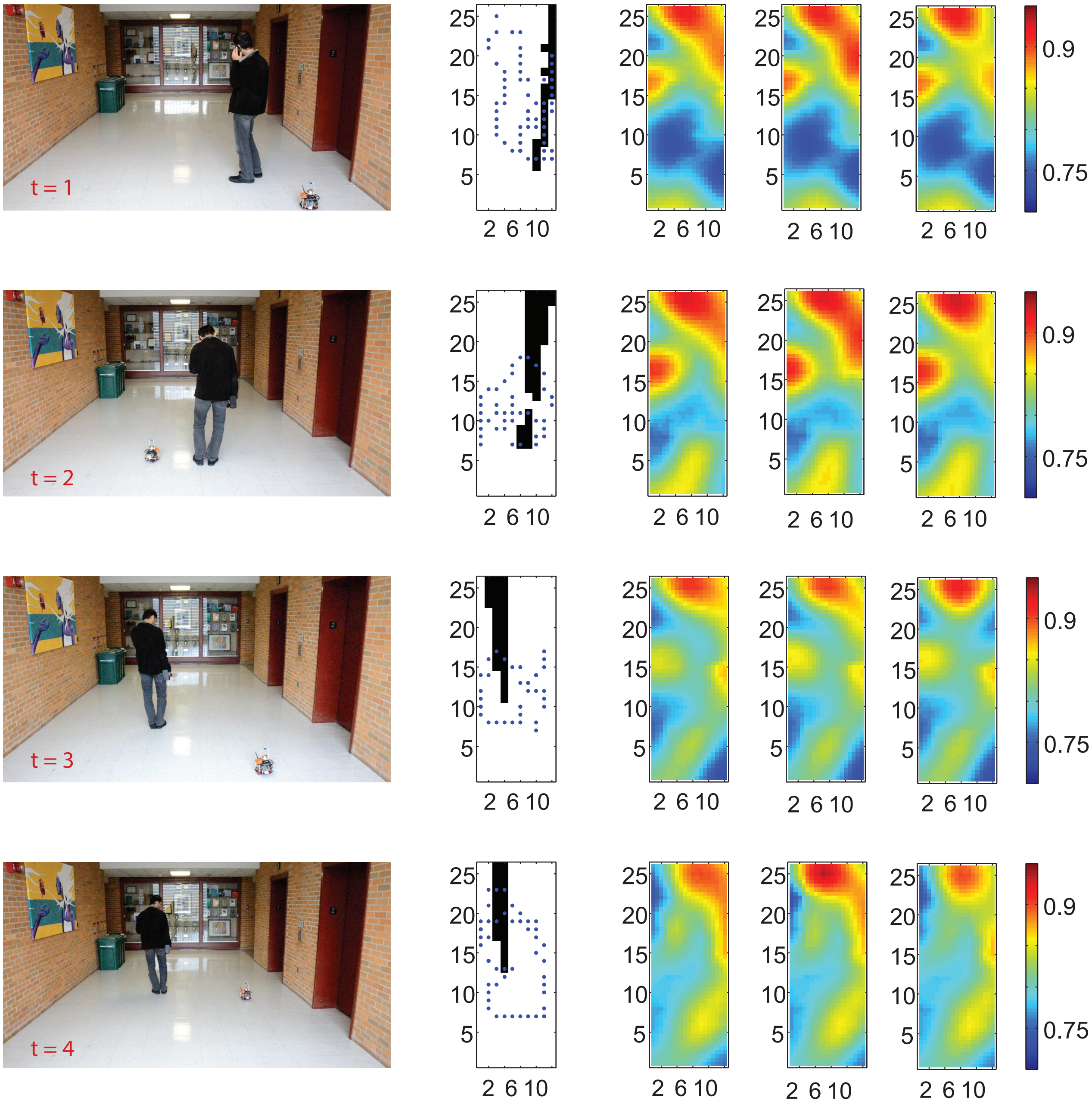


Figure 4.13: The time varying field is shown for 4 time iterations. The first column shows the moving object in the field. The second column shows the sampling positions with blue dots and the positioning blind spot with black areas. the third, forth and fifth columns show predicted field for Case 1, Case 3 and Case 4, respectively. The rows are correspond to the time intervals $t = 1, 2, 3$ and 4.

4.8 Conclusion

We have tackled a problem of predicting a spatio-temporal field using successive noisy measurements obtained by robotic sensors, some of which have uncertain localization. We developed the spatio-temporal field of interest using a GMRF and designed sequential prediction algorithms for computing the exact and approximated predictive inference from a Bayesian point of view. The most important contribution is that the computation times for Algorithm 3 and Algorithm 4 with a finite m at each time step do not grow as the number of measurements increases. Two different experimental results provide a solid proof of concept on the proposed scheme.

The proposed algorithm will be useful for robotics applications such as environmental monitoring by autonomous aquatic robots [40], decentralized environmental modeling by mobile sensor networks [64], gradient climbing of environmental fields [10,62] with or without localization uncertainty. Furthermore, our approach can provide a fully Bayesian way to automatically tune the hyperparameters in Gaussian process-based SLAM [7, 51, 77, 105] with increase computational complexity.

Chapter 5

Simultaneous Localization and Spatial Prediction

Abstract: This chapter investigates a fully Bayesian way to solve the simultaneous localization and spatial prediction (SLAP) problem using a Gaussian Markov random field (GMRF) model. The objective is to simultaneously localize robotic sensors and predict a spatial field of interest using sequentially obtained noisy observations collected by robotic sensors. The set of observations consists of the observed uncertain poses of robotic sensing vehicles and noisy measurements of a spatial field. To be flexible, the spatial field of interest is modeled by a GMRF with uncertain hyperparameters. We derive an approximate Bayesian solution to the problem of computing the predictive inferences of the GMRF and the localization, taking into account observations, uncertain hyperparameters, measurement noise, kinematics of robotic sensors, and uncertain localization. The effectiveness of the proposed algorithm is illustrated by simulation results.

5.1 Introduction

The simultaneous localization and mapping (SLAM) problem is important to be solved for a robot to explore an unknown environment under localization uncertainty [21]. The variations of the SLAM problem are surveyed and categorized with different perspectives in [97]. In general, most SLAM problems have strong geometric models [21,32,61,71,72,97]. For example, a robot learns the locations of the landmarks while localizing itself using triangulation algorithms. Such geometric models could be classified in two groups, viz., a sparse set of features which can be individually identified, often used in Kalman filtering methods [21], and a dense representation such as an occupancy grid, often used in particle filtering methods [31].

In contrast to the SLAM problem with popular geometrical models, there is a growing number of practical scenarios in which no such geometric model exists. Consider localization using various spatially distributed (continuous) signals such as distributed wireless Ethernet signal strength [27], or multi-dimensional magnetic fields [105]. Underwater autonomous gliders for ocean sampling cannot find usual geometrical models from measurements of environmental variables such as pH, salinity, and temperature [62]. Furthermore, there are reasons to avoid the geometric model as well, even when a geometric model does exist. Such cases may include: 1) the difficulty of reliably extracting sparse, stable features, 2) the ability to use all sensory data directly rather than a relatively small amount of abstracted discrete information obtained from feature extraction algorithms, and 3) high computational and storage costs of dealing with dense features.

Motivated by the aforementioned situations, in this chapter, we consider scenarios without geometric models and tackle the problem of simultaneous localization and prediction (SLAP) of a spatial field.

Nonparametric modeling and prediction techniques for random fields have been exploited for mobile robotic sensors [10,12,62,64,117]. Random fields such as Gaussian processes and Gaussian Markov random fields (GMRFs) [14, 84] have been frequently used for mobile sensor networks to statistically model physical phenomena such as harmful algal blooms, pH, salinity, temperature, and wireless signal strength e.g., [30, 57, 119].

The recent research efforts that are closely related to our problem are summarized as follows. In [43], the authors formulated Gaussian process regression under uncertain localization. In [42], the authors used a GMRF with uncertain hyperparameters and tackled a problem of prediction of the random field under localization uncertainty. However, kinematics or dynamics of the sensor vehicles were not incorporated in [42, 43]. In [7], Gaussian process regression was used to model geo-referenced sensor measurements (obtained from a camera). After training with data including noisy measurements and their exact sampling positions, a maximum likelihood estimator was used to find the best match for the location of each of newly sampled measurements. However, this was not SLAM since the training step has to be performed *a priori* for a given environment [7]. In [51, 105], Gaussian process regression was also used to implement SLAM using a magnetic field and the feasibility of such approaches were shown experimentally. The work in [77] used laser range-finder data to probabilistically classify the robot's environment into a region of occupancy. It provides a continuous representation of robot's surroundings by employing a Gaussian process. In [27], so called a WiF-SLAM problem was solved using a Gaussian process latent variable model (GP-LVM). However, the accurately known training data and the independence across the dimensions and instantiations of the data were assumed in [27], which may not be practical. To the best of our knowledge, most work related to our SLAP problem did not address uncertainties in the hyperparameters of the Gaussian process in a fully Bayesian way. In most of the previous work, the hyperparamters in the model were estimated offline *a priori*.

In this chapter, we formulate the simultaneous localization and spatial prediction (SLAP) problem, in order to simultaneously localize robotic sensors and predict a spatial random field of interest using sequentially obtained noisy observations collected by robotic sensors. The set of observations consists of the observed uncertain poses of robotic sensing vehicles and noisy measurements of a spatial field. To be flexible, the spatial field of interest is modeled by a GMRF with uncertain hyperparameters. We then derive an approximate Bayesian solution to the problem of computing the predictive inferences of the GMRF and the localization, taking into account observations, uncertain hyperparameters, measurement noise, kinematics of robotic sensors, and uncertain localization. The effectiveness of the proposed algorithm is illustrated by simulation results.

5.2 Sequential Bayesian inference with a GMRF

In this section, we define a GMRF model in detail, formulate the problem, and provide its solution.

5.2.1 Gaussian processes and Gaussian Markov random fields

In this section, we briefly introduce a GMRF as a discretized Gaussian process on a lattice. Consider a Gaussian process: $\mathbf{z} \sim \mathcal{GP}(\mu, \mathbf{K})$, where μ is the mean vector, and $\mathbf{K} \in \mathbb{R}^{n\times n}$ is the covariance matrix. We discretize the compact domain $\mathcal{S} := [0 \;\max(s_1)] \times [0 \;\max(s_2)]$ into n spatial sites $\mathcal{Q} := \left\{\mathring{s}^{(1)}, \cdots, \mathring{s}^{(n)}\right\}$, where $n = h\max(s_1) \times h\max(s_2)$. h will be chosen such that $n \in \mathbb{Z}_{>0}$. Note that $n \to \infty$ as $h \to \infty$. The collection of realized values of the random field in $\mathcal{Q}$ is denoted by $\mathbf{z} := (z^{(1)}, \cdots, z^{(n)})^T \in \mathbb{R}^n$, where $z^{(i)} := z(\mathring{s}^{(i)})$.

The prior distribution of $\mathbf{z}$ is given by $\mathcal{N}(\mu, \mathbf{K})$. We then have

$$\mathbb{P}(\mathbf{z}) \propto \exp\left(-\frac{1}{2}\left(\mathbf{z}-\mu\right)^T \mathbf{K}^{-1}\left(\mathbf{z}-\mu\right)\right). \tag{5.1}$$

The i, j-th element of $\mathbf{K}$ is defined as $(\mathbf{K})_{ij} = \mathbb{C}\mathrm{ov}(z^{(i)}, z^{(j)})$. The prior distribution of $\mathbf{z}$ can be written by a precision matrix $\mathbf{Q} = \mathbf{K}^{-1}$, i.e., $\mathbf{z} \sim \mathcal{N}(\mu, \mathbf{Q}^{-1})$. This can be viewed as a discretized version of the Gaussian process (or a GMRF) with a precision matrix $\mathbf{Q}$ on $\mathcal{Q}$. Note that $\mathbf{Q}$ of this GMRF is not sparse. However, a sparse version of $\mathbf{Q}$, i.e., $\hat{\mathbf{Q}}$ with local neighborhood that can represent the original Gaussian process can be found, for example, making $\hat{\mathbf{Q}}$ close to $\mathbf{Q}$ in some norm [15, 35, 90]. This approximate GMRF will be computationally efficient due to the sparsity of $\hat{\mathbf{Q}}$. In our simulation study, we will use a GMRF with a sparse precision matrix that represents a Gaussian process precisely as shown in [42, 63]. However, any parameterization of μ_θ and $\mathbf{Q}_\theta$, where θ is the hyperparameter vector, can be used.

5.2.2 Multiple robotic sensors

Consider N spatially distributed robots with sensors indexed by $j \in \mathcal{J} := \{1, \cdots, N\}$ sampling at time $t \in \mathbb{Z}_{>0}$. Suppose that the sampling time $t \in \mathbb{Z}_{>0}$ is discrete. Recall that the surveillance region is discretized as a lattice that consists of n spatial sites, whose set is denoted by $\mathcal{Q}$. Let n spatial sites in $\mathcal{Q}$ be indexed by $\mathcal{I} := \{1, \cdots, n\}$, and $\mathbf{z} := \mathrm{col}\left(z^{(1)}, \cdots, z^{(n)}\right) \in \mathbb{R}^n$ be the corresponding static values of the scalar field at n special sites. We denote all robots' locations at time t by $\mathbf{q}_t = \mathrm{col}\left(q_t^{[1]}, \cdots, q_t^{[N]}\right) \in \mathcal{Q}^N$, the observations made by all robots at time t by $\mathbf{y}_t = \mathrm{col}\left(y_t^{[1]}, \cdots, y_t^{[N]}\right) \in \mathbb{R}^N$, and the observed states of all robots at time t by $\tilde{\mathbf{s}}_t = \mathrm{col}\left(\tilde{s}_t^{[1]}, \cdots, \tilde{s}_t^{[N]}\right) \in \mathcal{S}^N$. $\tilde{\mathbf{s}}_t$ and $\mathbf{y}_t$ are noisy observations of $\mathbf{s}_t$ and $\mathbf{z}$, respectively. At time t, robot j takes a noise corrupted measurement at its current location $q_t^{[j]} = \mathring{s}^{(i)} \in \mathcal{Q}, \forall j \in \mathcal{J}, i \in \mathcal{I}$, viz.,

$$y_t^{[j]} = z^{(i)} + \epsilon_t^{[j]}, \tag{5.2}$$

where the measurement errors $\{\epsilon_t^{[j]}\}$ are assumed to be the independent and identically distributed (i.i.d.) Gaussian white noise, i.e., $\epsilon_t^{[j]} \overset{i.i.d.}{\sim} \mathcal{N}(0, {\sigma_\epsilon}^2)$. The measurement noise level ${\sigma_\epsilon}^2 > 0$ is assumed to be known, and we define $\boldsymbol{\epsilon}_t := \mathrm{col}\left(\epsilon_t^{[1]}, \cdots, \epsilon_t^{[N]}\right) \in \mathbb{R}^N$.

In addition, at time t, robot j takes a noisy observation of its own vehicle position.

$$\tilde{s}_t^{[j]} = q_t^{[j]} + e_t^{[j]}, \tag{5.3}$$

where the observation errors $\{e_t^{[j]}\}$ are distributed by $e_t^{[j]} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_e^2 I)$.

The observation noise level $\sigma_e^2 > 0$ is assumed to be known, and we define $\mathbf{e}_t := \mathrm{col}\left(e_t^{[1]}, \cdots, e_t^{[N]}\right) \in \mathbb{R}^{2N}$.

Our models can be represented in the concise collective notation.

$$\begin{aligned} \mathbf{y}_t &= H_t \mathbf{z} + \boldsymbol{\epsilon}_t, \\ \tilde{\mathbf{s}}_t &= L_t \mathbf{q}_t + \mathbf{e}_t, \end{aligned} \tag{5.4}$$

where L_t is the observation matrix for the vehicle states, and $H_\mathcal{T} \in \mathbb{R}^{n \times N}$ is defined by

$$(H_\mathcal{T})_{ij} = \begin{cases} 1, & \text{if } q_\mathcal{T}^{[j]} = \mathring{s}^{(i)}, \\ 0, & \text{otherwise.} \end{cases}$$

5.2.3 Kinematics of robotic vehicles

In this section, we introduce a specific model for the motion of robotic vehicles. Each robotic sensor is modeled by a nonholonomic differentially driven vehicle in a two dimensional domain, i.e., $\mathcal{S} \subset \mathbb{R}^2$. In this case, an equation of motion for robot i [86] may be given by

$$\begin{bmatrix} \dot{s}_{1,t}^{[i]} \\ \dot{s}_{2,t}^{[i]} \end{bmatrix} = \begin{bmatrix} u_t^{[i]} \cos \psi_t^{[i]} \\ u_t^{[i]} \sin \psi_t^{[i]} \end{bmatrix} + \beta_t^{[i]}, \tag{5.5}$$

where $\left\{s_{1,t}^{[i]}, s_{2,t}^{[i]}\right\}$, $\left\{\psi_t^{[i]}\right\}$, $\left\{u_t^{[i]}\right\}$, and $\beta_t^{[i]}$ denote the inertial position, the orientation, the linear speed, and the system noise of robot i in time t, respectively. In this case, the kinematics of the vehicle network can be further described in detail as follows.

$$\mathbf{s}_{t+1} = \mathbf{s}_t + F_t \mathbf{u}_t + \mathbf{v}_t, \tag{5.6}$$

where $\mathbf{u}_t$ is a known control input and $\mathbf{v}_t$ is an i.i.d. white noise realized by a known normal distribution $\mathcal{N}\left(0, \Sigma_{\mathbf{v}_t}\right)$.

The evolution of the location of the robot can be more detailed as follows.

$$\begin{aligned} \mathbf{q}_{t+1} &= \mathbb{Q}\left(\mathbf{q}_t + F_t \mathbf{u}_t + \mathbf{v}_t\right) \\ &= \mathbf{q}_t + F_t \mathbf{u}_t + \mathbf{w}_t, \end{aligned} \tag{5.7}$$

where $\mathbb{Q} : \mathcal{S} \to \mathcal{Q}$ is the nearest neighbor rule quantizer that takes an input and returns a projected value on $\mathcal{S}$. $\mathbf{v}_t$ is the process noise and $\mathbf{w}_t$ is the quantization error between the continuous and discretized states, i.e., $\mathbf{w}_t = \mathbb{Q}\left(\mathbf{q}_t + F_t \mathbf{u}_t + \mathbf{v}_t\right) - \left(\mathbf{q}_t + F_t \mathbf{u}_t\right)$. As the cardinality of $\mathcal{S}$ increases, we have that $\mathbf{w}_t \to \mathbf{v}_t$. A special case of (5.7) is that $F_t \mathbf{u}_t$ is controlled and $\mathbf{w}_t$ is chosen such that the next location $\mathbf{q}_{t+1}$ is on a grid point in $\mathcal{S}$. In this case, we have $\mathbf{v}_t = \mathbf{w}_t$.

Assuming that $\{\psi_t^{[i]} \,|\, \forall i \in \mathcal{J}\}$ in (5.5) can be measured precisly, $F_t \in \mathbb{R}^{2N \times N}$ in (5.5) is obtained as follows.

$$F_t = \Delta t \begin{bmatrix} \cos\psi_t^{[1]} & 0 & \cdots & 0 \\ \sin\psi_t^{[1]} & 0 & \cdots & 0 \\ 0 & \cos\psi_t^{[2]} & \cdots & 0 \\ 0 & \sin\psi_t^{[2]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cos\psi_t^{[N]} \\ 0 & 0 & \cdots & \sin\psi_t^{[N]} \end{bmatrix},$$

where $\Delta t = 1$ is a sampling time. We denote the collections of cumulative robots' locations, cumulative observations, and cumulative control inputs from time 1 to time t, respectively, by $\tilde{\mathbf{s}}_{1:t} := \text{col}\,(\tilde{\mathbf{s}}_1, \cdots, \tilde{\mathbf{s}}_t) \in \mathcal{S}^{Nt}$, $\mathbf{y}_{1:t} := \text{col}\,(\mathbf{y}_1, \cdots, \mathbf{y}_t) \in \mathbb{R}^{Nt}$, and $\mathbf{u}_{1:t} := \text{col}\,(\mathbf{u}_1, \cdots, \mathbf{u}_t) \in \mathbb{R}^{Nt}$.

5.2.4 Problem formulation and its Bayesian predictive inference

In this section, we formulate the SLAP problem and provide its Bayesian solution. To be precise, we present the following assumptions A.1-A.5 for the problem formulation.

A.1 The scalar random field $\mathbf{z}$ is generated by a GMRF model which is given by $\mathbf{z} \sim \mathcal{N}(\mu_\theta, \mathbf{Q}_\theta^{-1})$, where μ_θ and $\mathbf{Q}_\theta$ are given functions of a hyperparameter vector θ.

A.2 The noisy measurements $\{\mathbf{y}_t\}$ and the noisy sampling positions $\{\tilde{\mathbf{s}}_t\}$, as in (5.4), are collected by robotic sensors in time $t = 1, 2, \cdots$.

A.3 The control input $\{\mathbf{u}_t\}$ is a known deterministic vector at time t.

A.4 The prior distribution of the hyperparameter vector θ is discrete with a support $\Theta = \left\{\theta^{(1)}, \cdots, \theta^{(L)}\right\}$.

A.5 The prior distribution $\mathbb{P}(\mathbf{q}_t = \mathring{\mathbf{q}}_t^{(i)})$ is discrete with a support $\mathring{\mathbf{q}}_t^{(i)} \in \Omega(t)$, and

$$\Omega(t) = \left\{(\mathring{s}^{(k_1)}, \cdots, \mathring{s}^{(k_N)}) | (k_1, \cdots, k_N) \in \mathcal{L}(t)\right\}$$

, which is given at time t. Here, $\gamma(t) = |\mathcal{L}(t)|$ denotes the number of the probable possibilities for $\mathbf{q}_t$.

Problem 5.2.1. *Consider the assumptions A.1-A.5. Our problem is to simultaneously find the predictive distributions, means, and variances for both $\boldsymbol{z}$ and $\boldsymbol{q}$ conditional on $\mathcal{D}_t := \{\boldsymbol{y}_{1:t}, \tilde{\boldsymbol{s}}_{1:t}\}$.*

The solution to Problem 5.2.1 is derived as follows. The distribution of the GMRF is given by $\mathbb{P}\left(\mathbf{z}|\theta, \mathcal{D}_{t-1}\right) = \mathcal{N}\left(\mu_{\mathbf{z}|\theta,\mathcal{D}_{t-1}}, \Sigma_{\mathbf{z}|\theta,\mathcal{D}_{t-1}}\right)$. Recall that the evolution of $\mathbf{q}_t$ is given by (5.6) and the input $\mathbf{u}_t$ is a known deterministic vector at time t. Therefore, $\mathbb{P}\left(\mathbf{q}_t|\mathcal{D}_{t-1}\right)$ can be updated by the Gaussian approximation of $\mathbb{P}(\mathbf{q}_{t-1}|\mathcal{D}_{t-1})$.

$$\begin{aligned}\mathbb{P}\left(\mathbf{q}_t|\mathcal{D}_{t-1}\right) \propto \exp\Big((\mathbf{q}_t - \mu_{\mathbf{q}_{t-1}|\mathcal{D}_{t-1}} + F_{t-1}\mathbf{u}_{t-1})^T(\Sigma_{\mathbf{q}_{t-1}|\mathcal{D}_{t-1}} + \Sigma_{\mathbf{w}_{t-1}})^{-1} \\ \times\, (\mathbf{q}_t - \mu_{\mathbf{q}_{t-1}|\mathcal{D}_{t-1}} + F_{t-1}\mathbf{u}_{t-1})\Big).\end{aligned} \tag{5.8}$$

Similarly, $\mathbb{P}\left(\mathbf{y}_t|\theta, \mathcal{D}_{t-1}, \mathbf{q}_t\right)$ is updated by the Gaussian approximation of $\mathbb{P}(\mathbf{z}|\theta, \mathcal{D}_{t-1})$ as follows.

$$\mathbb{P}\left(\mathbf{y}_t|\theta, \mathcal{D}_{t-1}, \mathbf{q}_t\right) \approx \mathcal{N}\left(H_t^T \mu_{\mathbf{z}|\theta,\mathcal{D}_{t-1}}, \Sigma_{\epsilon_t} + H_t^T \Sigma_{\mathbf{z}|\theta,\mathcal{D}_{t-1}} H_t\right). \tag{5.9}$$

Remark 5.2.2. *For the sake of reducing memory usage and complexity, the distribution of* $\boldsymbol{q}_t|\mathcal{D}_{t-1}$ *and* $\boldsymbol{y}_t|\theta, \mathcal{D}_{t-1}, \boldsymbol{q}_t$ *are approximated by normal distributions.*

The joint distribution $\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_{t-1}$ is obtained as follows.

$$\mathbb{P}\left(\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_{t-1}\right) = \mathbb{P}\left(\mathbf{z}|\theta, \mathbf{q}_t, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\theta|\mathbf{q}_t, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\mathbf{q}_t|\mathcal{D}_{t-1}\right). \tag{5.10}$$

The observation model is given by (5.4), thus the probabilities of the observed data are $\mathbb{P}\left(\mathbf{y}_t|\mathbf{z}, \mathbf{q}_t\right) = \mathcal{N}\left(H_t^T \mathbf{z}, \Sigma_\epsilon\right)$ and $\mathbb{P}\left(\tilde{\mathbf{s}}_t|\mathbf{q}_t\right) = \mathcal{N}\left(L_t \mathbf{q}_t, \Sigma_{e_t}\right)$. The measured random variables have the following conditional joint distribution,

$$\mathbb{P}\left(\mathbf{y}_t, \tilde{\mathbf{s}}_t|\mathbf{z}, \mathbf{q}_t, \theta, \mathcal{D}_{t-1}\right) = \mathbb{P}\left(\mathbf{y}_t|\mathbf{z}, \theta, \mathbf{q}_t, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\tilde{\mathbf{s}}_t|\mathbf{q}_t\right). \tag{5.11}$$

From Bayes' rule, the posterior joint distribution of the scalar field values, the sampling positions, and the hyperparameter vector is given as follows.

$$\mathbb{P}\left(\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_t\right) = \frac{\mathbb{P}\left(\mathbf{y}_t, \tilde{\mathbf{s}}_t|\mathbf{z}, \mathbf{q}_t, \theta, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_{t-1}\right)}{\mathbb{P}\left(\mathbf{y}_t, \tilde{\mathbf{s}}_t|\mathcal{D}_{t-1}\right)}. \tag{5.12}$$

In addition, $\mathbb{P}\left(\mathbf{q}_t, \theta|\mathcal{D}_t\right) = \int \mathbb{P}\left(\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_t\right) d\mathbf{z}$ is given as follows.

$$\begin{aligned} \int \mathbb{P}\left(\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_t\right) d\mathbf{z} = & \frac{\mathbb{P}\left(\tilde{\mathbf{s}}_t|\mathbf{q}_t\right) \mathbb{P}\left(\theta|\mathbf{q}_t, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\mathbf{q}_t|\mathcal{D}_{t-1}\right)}{\mathbb{P}\left(\mathbf{y}_t, \tilde{\mathbf{s}}_t|\mathcal{D}_{t-1}\right)} \\ & \times \int \mathbb{P}\left(\mathbf{z}|\theta, \mathbf{q}_t, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\mathbf{y}_t|\mathbf{z}, \theta, \mathbf{q}_t, \mathcal{D}_{t-1}\right) d\mathbf{z}, \end{aligned} \tag{5.13}$$

where $\int \mathbb{P}\left(\mathbf{z}|\theta, \mathbf{q}_t, \mathcal{D}_{t-1}\right) \mathbb{P}\left(\mathbf{y}_t|\mathbf{z}, \theta, \mathbf{q}_t, \mathcal{D}_{t-1}\right) d\mathbf{z} = \mathbb{P}\left(\mathbf{y}_t|\mathbf{q}_t, \theta, \mathcal{D}_{t-1}\right)$, and $\mathbb{P}(\mathbf{y}_t|\mathbf{q}_t, \theta, \mathcal{D}_{t-1})$ is given by (5.9).

Remark 5.2.3. *From the Bayes' rule,* $\mathbb{P}(\theta|\boldsymbol{q}_t, \mathcal{D}_{t-1})$ *is given by* $\frac{\mathbb{P}(\theta, \boldsymbol{q}_t|\mathcal{D}_{t-1})}{\mathbb{P}(\boldsymbol{q}_t|\mathcal{D}_{t-1})}$. *We can* *compute* $\mathbb{P}(\theta, \boldsymbol{q}_t|\mathcal{D}_{t-1})$ *for all the possible combinations of* $\boldsymbol{q}_t$ *in the previous iteration using* (5.13). *However, for the sake of reducing the computational cost, we approximate* $\mathbb{P}(\theta|\boldsymbol{q}_t, \mathcal{D}_{t-1})$ *by* $\mathbb{P}(\theta|\mathcal{D}_{t-1})$. *Therefore, we have*

$$\mathbb{P}(\boldsymbol{q}_t, \theta|\mathcal{D}_t) \approx \frac{\mathbb{P}(\tilde{\boldsymbol{s}}_t|\boldsymbol{q}_t)\,\mathbb{P}(\theta|\mathcal{D}_{t-1})\,\mathbb{P}(\boldsymbol{q}_t|\mathcal{D}_{t-1})\,\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{q}_t, \theta, \mathcal{D}_{t-1})}{\mathbb{P}(\boldsymbol{y}_t, \tilde{\boldsymbol{s}}_t|\mathcal{D}_{t-1})},$$

Marginalizing out uncertainties on the possible $\mathbf{q}_t$ and θ, we obtain the following.

$$\begin{aligned}\mathbb{P}(\mathbf{z}, \theta|\mathcal{D}_t) &= \sum_{\mathbf{q}_t \in \Omega(t)} \mathbb{P}(\mathbf{z}, \mathbf{q}_t, \theta|\mathcal{D}_t),\\ \mathbb{P}(\mathbf{z}|\mathcal{D}_t) &= \sum_{\theta \in \Theta} \mathbb{P}(\mathbf{z}, \theta|\mathcal{D}_t).\end{aligned} \tag{5.14}$$

Our estimation of $\mathbf{q}_t$ and θ can be corrected using measured data up to time t as follows.

$$\begin{aligned}\mathbb{P}(\mathbf{q}_t|\mathcal{D}_t) &= \sum_{\theta \in \Theta} \mathbb{P}(\mathbf{q}_t, \theta|\mathcal{D}_t),\\ \mathbb{P}(\theta|\mathcal{D}_t) &= \sum_{\mathbf{q}_t \in \Omega(t)} \mathbb{P}(\mathbf{q}_t, \theta|\mathcal{D}_t).\end{aligned} \tag{5.15}$$

The predictive probability and the mean value of $\mathbf{z}|\theta, \mathcal{D}_t$ are obtained as follows.

$$\begin{aligned}\mathbb{P}(\mathbf{z}|\theta, \mathcal{D}_t) &= \frac{\mathbb{P}(\mathbf{z}, \theta|\mathcal{D}_t)}{\mathbb{P}(\theta|\mathcal{D}_t)},\\ \mu_{\mathbf{z}|\theta,\mathcal{D}_t} &= \frac{1}{\mathbb{P}(\theta|\mathcal{D}_t)} \sum_{\mathbf{q}_t \in \Omega(t)} \mu_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t}\,\mathbb{P}(\mathbf{q}_t, \theta|\mathcal{D}_t)\,.\end{aligned} \tag{5.16}$$

The predictive covariance matrix of $\mathbf{z}|\theta, \mathcal{D}_t$ can be obtained using the law of total variance $\Sigma_{\mathbf{z}|\theta,\mathcal{D}_t} = \mathbb{E}\left(\Sigma_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t}\right) + \mathbb{C}\mathrm{ov}\left(\mu_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t}\right)$, where the $\mathbb{E}$ and $\mathbb{C}$ov is computed over random variable $\mathbf{q}_t$. Such variables are obtained as follows.

$$\begin{aligned}
\mathbb{E}\left(\Sigma_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t}\right) &= \sum_{\mathbf{q}_t \in \Omega(t)} \Sigma_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t} \mathbb{P}\left(\mathbf{q}_t|\theta,\mathcal{D}_t\right), \\
\mathbb{C}\mathrm{ov}\left(\mu_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t}\right) &= \sum_{\mathbf{q}_t \in \Omega(t)} \left(\mu_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t} - \mu_{\mathbf{z}|\theta,\mathcal{D}_t}\right)\left(\mu_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t} - \mu_{\mathbf{z}|\theta,\mathcal{D}_t}\right)^T \mathbb{P}\left(\mathbf{q}_t|\theta,\mathcal{D}_t\right),
\end{aligned} \tag{5.17}$$

where the predictive mean and covariance of $\mathbf{z}|\mathbf{q}_t, \theta, \mathcal{D}_t$ are calculated using Gaussian process regression as follows.

$$\begin{aligned}
\mu_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t} =& \mu_{\mathbf{z}|\theta,\mathcal{D}_{t-1}} \quad + \Sigma_{\mathbf{z}|\theta,\mathcal{D}_{t-1}} H_{\mathbf{q}_t}^T \Sigma_{\mathbf{y}_t|\theta,\mathcal{D}_{t-1},\mathbf{q}_t}^{-1} (\mathbf{y}_t - \mu_{\mathbf{y}_t|\theta,\mathcal{D}_{t-1},\mathbf{q}_t}), \\
\Sigma_{\mathbf{z}|\mathbf{q}_t,\theta,\mathcal{D}_t} =& \Sigma_{\mathbf{z}|\theta,\mathcal{D}_{t-1}} - \Sigma_{\mathbf{z}|\theta,\mathcal{D}_{t-1}} H_{\mathbf{q}_t}^T \Sigma_{\mathbf{y}_t|\theta,\mathcal{D}_{t-1},\mathbf{q}_t}^{-1} H_{\mathbf{q}_t} \Sigma_{\mathbf{z}|\theta,\mathcal{D}_{t-1}}.
\end{aligned} \tag{5.18}$$

Finally, the first and second moments of $\mathbf{q}_t|\mathcal{D}_t$ are obtained as follows.

$$\begin{aligned}
\mu_{\mathbf{q}_t|\mathcal{D}_t} &= \sum_{\mathbf{q}_t \in \Omega(t)} \mathbf{q}_t \, \mathbb{P}\left(\mathbf{q}_t|\mathcal{D}_t\right), \\
\Sigma_{\mathbf{q}_t|\mathcal{D}_t} &= \sum_{\mathbf{q}_t \in \Omega(t)} \left(\mathbf{q}_t - \mu_{\mathbf{q}_t|\mathcal{D}_t}\right)^2 \mathbb{P}\left(\mathbf{q}_t|\mathcal{D}_t\right).
\end{aligned}$$

5.3 Simulation Results

In this section, we demonstrate the effectiveness of the proposed sequential Bayesian inference algorithm using a numerical experiment. Consider a robot is moving in a discretized surveillance region $\mathcal{Q}$. The spatial sites in $\mathcal{Q}$ consist of 31×31 grid points, i.e., $n = 961$, uniformly distributed over the surveillance region $\mathcal{S} := [-15, 15] \times [-15, 15]$.

In this illustrative example, we realize the spatial field developed in [42], which a GMRF wrapped around in a torus structure. Thus the top edge (respectively, the left edge) and the bottom edge (respectively, the right edge) are neighbors each other. The parameters of the model in [42] are selected as follows. The mean vector μ_θ is chosen to be zero, and the precision matrix Q_θ is chosen with hyperparameters $\alpha = 0.1$ equivalent to a bandwidth $\ell = \sqrt{2}/\sqrt{\alpha} \approx 4.47$, and $\kappa = 50$ equivalent to $\sigma_f^2 = 1/4\pi\alpha\kappa \approx 0.016$. The prior distribution of the hyperparameter vector θ is discrete with a support

$$\Theta = \{(\kappa, \alpha), (0.1\kappa, \alpha), (10\kappa, \alpha), (\kappa, 0.1\alpha), (\kappa, 10\alpha)\},$$

along with the corresponding uniform probabilities $\{0.2, 0.2, 0.2, 0.2, 0.2\}$. The measurement noise variance in (5.2) is given by $\sigma_\epsilon = 0.1$.

A robot takes measurements at time $t \in \{1, 2, \cdots, 100\}$ with localization uncertainty. In Figs. 5.1-(d), (e), and (f), true, noisy, and probable sampling positions are shown in circles, stars, and dots, respectively, at time $t = 100$. In this simulation, the standard deviation of the noise in the observed sampling position is given by $\sigma_e = 10$ in (5.3). The probable sampling positions that form support $\Omega(t)$, are selected within the confidence region of $\Pr(\|\mathring{s}_t^{(i)} - \tilde{q}_t^{[j]}\| \leq \sigma_e)$.

The results of the simultaneous localization and spatial prediction are summarized for three methods as follows.

- *Case 1:* Figs. 5.1-(a), (d), and (g) show the prediction, prediction error variance, and squared (empirical) error fields, using exact sampling positions. With the true sampling positions, the best prediction quality is expected for this case.

- *Case 2:* Figs. 5.1-(b), (e), and (h) show the results, by using sampled noisy positions. The results clearly illustrate that naively applying GMRF regression to noisy sampling positions can potentially distort prediction at a significant level. Fig. 5.1-(h) shows that squared error of this case is considerably higher than that of Case 1.

- *Case 3:* Figs. 5.1-(c), (f), and (i) show the results, by applying the proposed approach in Section 5.2.4. The resulting prediction quality is much improved as compared to Case 2 and is even comparable to that of Case 1.

The averaged squared errors in time and space, using true sampling positions (Case 1), noisy sampling positions (Case 2), and using uncertain sampling positions (Case 3) are $0.0837 \times 10^{-3}, 0.1664 \times 10^{-3}$, and 0.0989×10^{-3}, respectively. This shows the effectiveness of our solution to Problem 5.2.1.

The true positions of the robot for time $t \in \mathcal{T} := \{10, 31, \cdots, 30\}$ are shown in Fig. 5.2 by red diamonds and lines. The estimated sampling positions of the robot $\mathbb{E}(\mathbf{q}_t|\mathcal{D}_t)$ for $t \in \mathcal{T}$ are shown in blue dots with estimated confidence regions. Fig. 5.2 clearly shows that the proposed approach in this chapter significantly reduce the localization uncertainty as compared to the noise level of the sampled positions (denoted by green stars).

In this example, the fixed running time using Matlab R2009b (MathWorks) on a PC (3.2 GHz Intel i7 Processor) is about 40 seconds for each iteration of time which is fast enough for real world implementation.

5.4 Conclusion

In this chapter, we provide an approximate Bayesian solution to the problem of simultaneous localization and spatial prediction (SLAP), taking into account kinematics of robots and uncertainties in the precision matrix, the sampling positions, and the measurements of a GMRF in a fully Bayesian manner. In contrast to [42], the kinematics of the robotic vehicles

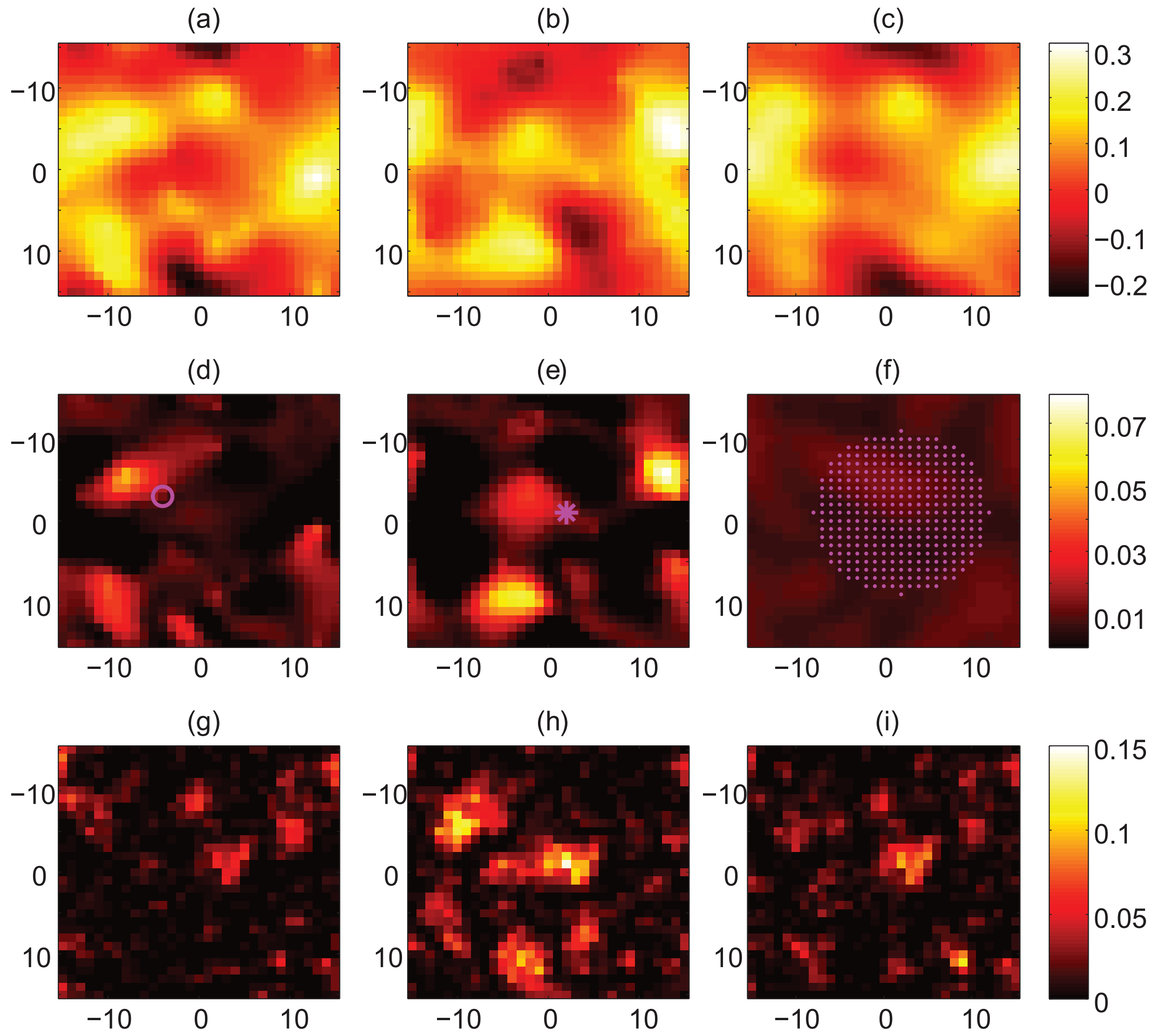


Figure 5.1: The prediction results of Cases 1, 2, and 3 at time $t = 100$ are shown in the first, second, third columns, respectively. The first, second, and third rows correspond to the prediction, prediction error variance, and squared empirical error fields between the prediction and true fields. True, noisy, and probable sampling positions are shown in circles, stars, and dots, respectively. The x and y axis represent 2-D localization, and colors represent the value of the desired quantity in the locations.

are integrated into the inference algorithm. The simulation results show that the proposed approach estimates the sampling positions and predicts the spatial field along with their prediction error variances successfully, in a fixed computational time. The simulation study suggests that the complexity of the proposed scalable inference algorithm is affordable for a

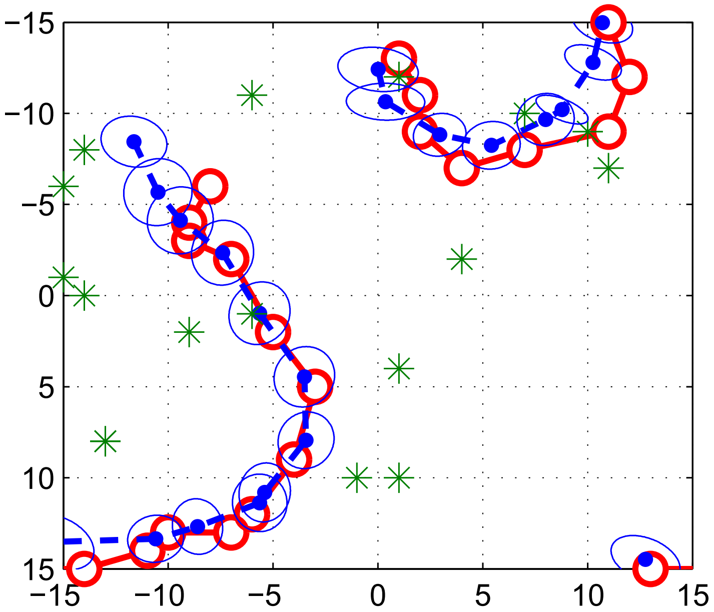


Figure 5.2: The trajectories of true, predicted, and noisy sampling positions of the robot are shown by red diamonds, blue dots, and green stars for time $t \in \{10, 11, \cdots, 30\}$. The blue ellipsoids show the confidence regions of about 68% for the estimated sampling positions.

robot to operate in real world situations.

Chapter 6

Conclusion and Future Work

In this chapter, we briefly summarize the key contributions presented in this dissertation and propose some directions for future work.

6.1 Conclusion

In Chapter 2, we have formulated Gaussian process regression with observations under the localization uncertainty. Effects of the measurements noise, localization uncertainty, and prior distributions have been all correctly incorporated in the posterior predictive statistics in a Bayesian approach. We have reviewed the Monte Carlo sampling and Laplace's method, which have been applied to compute the analytically intractable posterior predictive statistics of the Gaussian processes with localization uncertainty. The approximation error and complexity of all the proposed approaches have been analyzed. In particular, we have provided tradeoffs between the error and complexity of Laplace approximations and their different degrees such that one can choose a tradeoff taking into account the performance requirements and computation complexity due to the resource-constrained sensor network. Simulation study demonstrated that the proposed approaches perform much better than approaches without considering the localization uncertainty properly.

In Chapter 3, the multivariate Gaussian process model is utilized to use vision data for the mobile robot localization. The predictive statistics of vision data is learned in advance and used in order to estimate the position of a mobile robot, equipped just with an omnidirectional camera. For the sake of flexibility, the multivariate GP model with unknown hyperparameters is used to model vision data. The experiments show good positioning results within a reasonable computational time.

In Chapter 4, the problem of predicting a spatio-temporal field using successive noisy measurements is tackled. We developed the spatio-temporal field of interest using a GMRF and designed sequential prediction algorithms for computing the exact and approximated predictive inference from a Bayesian point of view. The most important contribution is that the computation times for the proposed algorithms do not grow as the number of measurements increases.

In Chapter 5, we provide an approximate Bayesian solution to the problem of simultaneous localization and spatial prediction (SLAP), taking into account kinematics of robots and uncertainties in the precision matrix, the sampling positions, and the measurements of a GMRF in a fully Bayesian manner. In contrast to Chapter 4, the kinematics of the robotic vehicles are integrated into the inference algorithm. The simulation results show that the proposed approach estimates the sampling positions and predicts the spatial field along with their prediction error variances successfully, in a fixed computational time. The simulation study suggests that the complexity of the proposed scalable inference algorithm is affordable for a robot to operate in real world situations.

6.2 Future Work

In the long term, we plan to expanding our current work and exploring on the following directions.

Although optimal sampling was not pursued in this dissertation, it will be interesting to consider the optimal sampling strategies for mobile sensing agents subject to complicated vehicle dynamics.

The distributed implementation of the proposed algorithms is practically important, especially for a sensor network with a limited computational power. To this end, we are interested in investigating on the efficient implementation and emerging technologies such as GPU computing to speed up current algorithms for real time applications.

Expanding the work on spatial modeling using GMRFs to deal with a general spatio-temporal process is an important direction to follow in the future. Implementing the developed algorithms in more challenging and more realistic experiments will play a key role in expanding the scope of applications of the proposed methods in this dissertation.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Xsens company, MTi-G, miniature AHRS with integreted GPS, 2009.

[2] C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.

[3] C. M. Bishop. *Pattern recognition and machine learning.* Springer, New York, 2006.

[4] P. Blaer and P. Allen. Topological mobile robot localization using fast vision techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation,*, volume 1, pages 1031–1036, 2002.

[5] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.

[6] A. Brix and P.J. Diggle. Spatiotemporal prediction for log-Gaussian Cox processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4):823–841, 2001.

[7] A. Brooks, A. Makarenko, and B. Upcroft. Gaussian process models for indoor and outdoor sensor-centric robot localization. *IEEE Transactions on Robotics*, 24(6):1341–1351, 2008.

[8] J. Choi, J. Lee, and S. Oh. Biologically-inspired navigation strategies for swarm intelligence using spatial Gaussian processes. In *Proceedings of the 17th International Federation of Automatic Control World Congress*, pages 593–598, July 2008.

[9] J. Choi, J. Lee, and S. Oh. Swarm intelligence for achieving the global maximum using spatio-temporal Gaussian processes. In *Proceedings of the 27th American Control Conference*, pages 135–140, June 2008.

[10] J. Choi, S. Oh, and R. Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802–2814, December 2009.

[11] G.C. Christie and C.I. Goddard. Sea lamprey international symposium (slis ii): advances in the integrated management of sea lamprey in the great lakes. *Journal of Great Lakes Research*, 29:1–14, 2003.

[12] J. Cortés. Distributed Kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827, December 2009.

[13] J. Cortés. Distributed Kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827, 2010.

[14] N. Cressie. Kriging nonstationary data. *Journal of the American Statistical Association*, 81(395):625–634, September 1986.

[15] N. Cressie and N. Verzelen. Conditional-mean least-squares fitting of Gaussian Markov random fields to Gaussian fields. *Computational Statistics & Data Analysis*, 52(5):2794–2807, January 2008.

[16] N. Cressie and C.K. Wikle. Space–time Kalman filter. *Encyclopedia of environmetrics*, 4:2045–2049, 2002.

[17] D. Culler, D. Estrin, and M. Srivastava. Guest editors' introduction: overview of sensor networks. *Computer*, 37(8):41–49, August 2004.

[18] M.P. Deisenroth, C.E. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 72(7-9):1508–1524, 2009.

[19] Cédric Demonceaux, Pascal Vasseur, and Yohan Fougerolle. Central catadioptric image processing with geodesic metric. *Image and Vision Computing*, 29(12):840–849, 2011.

[20] Guilherme N. DeSouza and Avinash C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 24(2):237–267, 2002.

[21] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[22] Vladislavs Dovgalecs, Rémi Mégret, and Yannick Berthoumieu. Multiple feature fusion based on co-training approach and time regularization for place classification in wearable video. *Advances in Multimedia*, 1, 2013.

[23] Y. Dupuis, X. Savatier, and P. Vasseur. Feature subset selection applied to model-free gait recognition. *Image and Vision Computing*, 31(8):580 – 591, 2013.

[24] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, 2006.

[25] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *Pervasive computing*, 1(1):59–69, 2002.

[26] João Carlos Aparício Fernandes and José Alberto B Campos Neves. Using conical and spherical mirrors with conventional cameras for 360 panorama views in a single image. In *Proceedings of the IEEE International Conference on Mechatronics*, pages 157–160, 2006.

[27] B. Ferris, D. Fox, and N. Lawrence. WiFi-SLAM using Gaussian process latent variable models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2480–2485, 2007.

[28] J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, 57(6):1317–1339, 1989.

[29] A. Girard, C.E. Rasmussen, J.Q. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting. *Advances in Neural Information Processing Systems*, pages 545–552, 2003.

[30] R. Graham and J. Cortés. Cooperative adaptive sampling of random fields with partially known covariance. *International Journal of Robust and Nonlinear Control*, 22(5):504–534, March 2012.

[31] G. Grisettiyz, C. Stachniss, and W. Burgard. Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2432–2437. IEEE, 2005.

[32] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[33] Baofeng Guo and Mark S Nixon. Gait feature subset selection by mutual information. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(1):36–46, 2009.

[34] Efstathios Hadjidemetriou, Michael D. Grossberg, and Shree K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, 2004.

[35] L. Hartman and O. Hössjer. Fast kriging of large data sets with Gaussian Markov random fields. *Computational Statistics & Data Analysis*, 52(5):2331–2349, January 2008.

[36] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

[37] KC Ho. Alleviating sensor position error in source localization using calibration emitters at inaccurate locations. *IEEE Transactions on Signal Processing*, 58(1):67–83, 2010.

[38] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on mobile computing and networking*, pages 45–57. ACM, 2004.

[39] M. Jadaliha and J. Choi. Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments. *IEEE Transactions on Control Systems Technology.* in press, 2012.

[40] M. Jadaliha and J. Choi. Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments. *IEEE Transactions on Control Systems Technology*, 21(3):899–905, 2013.

[41] M. Jadaliha, Y. Xu, , and J. Choi. Efficient Spatial Prediction Using Gaussian Markov Random Fields under Uncertain Localization. In *Proceedings of the ASME Dynamic Systems and Control Conference, October 17-19, 2012, Ft. Lauderdale, FL, USA.*, pages 1–6, 2012.

[42] M. Jadaliha, Y. Xu, and J. Choi. Efficient Spatial Prediction Using Gaussian Markov Random Fields Under Uncertain Localization. In *Proceeding of the ASME Dynamic Systems and Control Conference*, October 2012.

[43] M. Jadaliha, Y. Xu, and J. Choi. Gaussian process regression using Laplace approximations under localization uncertainty. In *Proceedings of the American Control Conference*, June 2012.

[44] Mahdi Jadaliha, Yunfei Xu, Jongeun Choi, N Johnson, and Weiming Li. Gaussian process regression for sensor networks under localization uncertainty. *IEEE Transactions on Signal Processing*, 61(2):223–237, 2013.

[45] Gijeong Jang, Sungho Lee, and Inso Kweon. Color landmark based self-localization for indoor mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 1037–1042, 2002.

[46] Woo Yeon Jeong and Kyoung Mu Lee. Visual slam with line and corner features. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2570–2575. IEEE, 2006.

[47] L. Jetto, S. Longhi, and G. Venturini. Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15(2):219–229, April 1999.

[48] N.S. Johnson, M.A. Luehring, M.J. Siefkes, and W. Li. Mating pheromone reception and induced behavior in ovulating female sea lampreys. *North American journal of fisheries management*, 26(1):88–96, 2006.

[49] N.S. Johnson, S.S. Yun, H.T. Thompson, C.O. Brant, and W. Li. A synthesized pheromone induces upstream movement in female sea lamprey and summons them into traps. *Proceedings of the National Academy of Sciences*, 106(4):1021, 2009.

[50] R. Karlsson and F. Gustafsson. Bayesian surface and underwater navigation. *IEEE Transactions on Signal Processing*, 54(11):4204–4213, 2006.

[51] A. Kemppainen, J. Haverinen, I. Vallivaara, and J. Röning. Near-optimal SLAM exploration in Gaussian processes. In *Proceeding of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 7–13, September 2010.

[52] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745, 2009.

[53] J. Ko and D. Fox. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90, 2009.

[54] J. Kocijan, R. Murray-Smith, C.E. Rasmussen, and B. Likar. Predictive control with Gaussian process models. In *Proceedings of EUROCON 2003*, volume 1. IEEE, 2003.

[55] J. Kosecka, Liang Zhou, P. Barber, and Z. Duric. Qualitative image based localization in indoors environments. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–3–II–8, 2003.

[56] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 2–10. ACM, 2006.

[57] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[58] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems*, pages 625–632, 2003.

[59] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[60] J. Le Ny and G. J. Pappas. On trajectory optimization for active sensing in Gaussian process models. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 6286–6292, December 2009.

[61] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceeding of the Intelligent Robots and Systems (IROS)*, pages 1442–1447. IEEE, 1991.

[62] N.E. Leonard, D.A. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[63] F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B*, 73(4):423–498, September 2011.

[64] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, June 2008.

[65] A. McHutchon and C.E. Rasmussen. Gaussian Process training with input noise. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS), Grandata, Spain,*, pages 1–9, December 2011.

[66] Emanuele Menegatti, Takeshi Maeda, and Hiroshi Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[67] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. Image-based Monte Carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[68] Y. Miyata. Fully exponential Laplace approximations using asymptotic modes. *Journal of the American Statistical Association*, 99(468):1037–1049, 2004.

[69] Y. Miyata. Laplace approximations to means and variances with asymptotic modes. *Journal of Statistical Planning and Inference*, 140(2):382–392, 2010.

[70] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

[71] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceeding of the Artificial Intelligence*, volume 18, pages 1151–1156, 2003.

[72] K. Murphy. Bayesian map learning in dynamic environments. *Advances in Neural Information Processing Systems (NIPS)*, 12:1015–1021, 1999.

[73] R. Murray-Smith, D. Sbarbaro, C.E. Rasmussen, and A. Girard. Adaptive, cautious, predictive control with Gaussian process priors. In *Proceeding of the 13th IFAC Symposium on System Identification, Rotterdam, Netherlands*, pages 1195–1200, 2003.

[74] M.F. Mysorewala, D.O. Popa, and F.L. Lewis. Multi-scale adaptive sampling with mobile agents for mapping of forest fires. *Journal of Intelligent and Robotic Systems*, 54(4):535–565, 2009.

[75] D. Nash and M. Hannah. Using Monte-Carlo simulations and Bayesian Networks to quantify and demonstrate the impact of fertiliser best management practices. *Environmental Modelling & Software*, 26(9):1079–1088, 2011.

[76] Mark Nixon and Alberto S Aguado. *Feature extraction & image processing*. Academic Press, 2008.

[77] S. O'Callaghan, F.T. Ramos, and H. Durrant-Whyte. Contextual occupancy maps using Gaussian processes. In *Proceeding of the IEEE International Conference on Robotics and Automation*, pages 1054–1060, May 2009.

[78] P. Oguz-Ekim, JP Gomes, J. Xavier, and P. Oliveira. Robust localization of nodes and time-recursive tracking in sensor networks using noisy range measurements. *IEEE Transactions on Signal Processing*, 59(99):3930–3942, 2011.

[79] S. Oh, L. Schenato, P. Chen, and S. Sastry. Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments. *Proceedings of the IEEE*, 95(1):234–254, January 2007.

[80] Naoya Ohnishi and Atsushi Imiya. Appearance-based navigation and homing for autonomous mobile robot. *Image and Vision Computing*, 31(6):511 – 532, 2013.

[81] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[82] Christian Plagemann, Cyrill Stachniss, Jrgen Hess, Felix Endres, and Nathan Franklin. A nonparametric learning approach to range sensing from omnidirectional vision. *Robotics and Autonomous Systems*, 58(6):762 – 772, 2010.

[83] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[84] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, Cambridge, Massachusetts, London, England, 2006.

[85] Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

[86] W. Ren. Consensus strategies for cooperative control of vehicle formations. *Control Theory & Applications, IET*, 1(2):505–512, 2007.

[87] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, 2007.

[88] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. Chapman & Hall, 2005.

[89] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, April 2009.

[90] H. Rue and H. Tjelmeland. Fitting Gaussian Markov random fields to Gaussian fields. *Scandinavian Journal of Statistics*, 29(1):31–49, March 2002.

[91] Timo Schairer, Benjamin Huhle, Philipp Vorst, Andreas Schilling, and Wolfgang Straser. Visual mapping with uncertainty for correspondence-free localization using Gaussian process regression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4229–4235, 2011.

[92] D Scharstein and A.J Briggs. Real-time recognition of self-similar landmarks. *Image and Vision Computing*, 19(11):763 – 772, 2001.

[93] Matthias Seeger. *Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations*. PhD thesis, School of Informatics, University of Edinburgh, 2003.

[94] BR Smith and JJ Tibbles. Sea lamprey (petromyzon marinus) in lakes huron, michigan, and superior: history of invasion and control, 1936-78. *Canadian Journal of Fisheries and Aquatic Sciences*, 37(11):1780–1801, 1980.

[95] A. J. Smola and P. L. Bartlett. Sparse greedy Gaussian process regression. In *Proceeding of the Advances in Neural Information Processing Systems 13*, pages 619–625, 2001.

[96] M. St-Pierre and D. Gingras. Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In *Proceeding of the IEEE Intelligent Vehicles Symposium*, pages 831–835. IEEE, June 2004.

[97] S. Thrun. Simultaneous localization and mapping. *Robotics and cognitive approaches to spatial mapping*, pages 13–41, 2008.

[98] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141, May 2001.

[99] Sebastian Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, 1998.

[100] L. Tierney and J.B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

[101] L. Tierney, R.E. Kass, and J.B. Kadane. Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association*, 84(407):710–716, 1989.

[102] D.H. Titterton and J.L. Weston. *Strapdown inertial navigation technology.* Peter Peregrinus Ltd, 2004.

[103] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 273–280, 2003.

[104] Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

[105] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning. Simultaneous localization and mapping using ambient magnetic field. In *Proceeding of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 14–19, September 2010.

[106] Shrihari Vasudevan. Data fusion with Gaussian processes. *Robotics and Autonomous Systems*, 60(12):1528 – 1544, 2012.

[107] N. Vlassis, R. Bunschoten, and B. Krose. Learning task-relevant features from robot data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 499–504, 2001.

[108] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[109] Gordon Wells and Carme Torras. Assessing image features for vision-based robot positioning. *Journal of Intelligent and Robotic Systems*, 30(1):95–118, 2001.

[110] Gordon Wells, Christophe Venaille, and Carme Torras. Vision-based robot positioning using neural networks. *Image and Vision Computing*, 14(10):715–732, 1996.

[111] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8:514–520, 1996.

[112] Christopher Williams and Matthias Seeger. Using the Nystrom method to speed up kernel machines. In *Proceeding of the Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[113] Y. Xu and J. Choi. Adaptive sampling for learning Gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066, 2011.

[114] Y. Xu and J. Choi. Adaptive sampling for learning Gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066, March 2011.

[115] Y. Xu, J. Choi, S. Dass, and T. Maiti. Bayesian prediction and adaptive sampling algorithms for mobile sensor networks. In *Proceedings of the 2011 American Control Conference*, pages 4195–4200, June 2011.

[116] Y. Xu, J. Choi, S. Dass, and T. Maiti. Efficient Bayesian spatial prediction with mobile sensor networks using Gaussian Markov random fields. 2012. submitted to Automatica.

[117] Y. Xu, J. Choi, S. Dass, and T. Maiti. Sequential Bayesian prediction and adaptive sampling algorithms for mobile sensor networks. *IEEE Transactions on Automatic Control*, 57(8):2078–2084, 2012.

[118] Y. Xu, J. Choi, and S. Oh. Mobile sensor network coordination using Gaussian processes with truncated observations. *IEEE Transactions on Robotics*, 27(6):1118–1131, 2011.

[119] Y. Xu, J. Choi, and S. Oh. Mobile sensor network navigation using Gaussian processes with truncated observations. *IEEE Transactions on Robotics*, 27(6):1118 – 1131, December 2011.

[120] Yunfei Xu and Jongeun Choi. Spatial prediction with mobile sensor networks using Gaussian processes with built-in Gaussian Markov random fields. *Automatica*, 48(8):1735 – 1740, 2012.