Fully Bayesian Simultaneous Localization and Spatial Prediction using Gaussian Markov Random Fields (GMRFs)

Mahdi Jadaliha, Do Huan and Jongeun Choi

Abstract— **This paper 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.**

I. Introduction

The simultaneous localization and mapping (SLAM) problem is important to be solved for a robot to explore an unknown environment under localization uncertainty [1]. The variations of the SLAM problem are surveyed and categorized with different perspectives in [2]. In general, most SLAM problems have strong geometric models [1]–[6]. 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 [1], and a dense representation such as an occupancy grid, often used in particle filtering methods [7].

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 [8], or multi-dimensional magnetic fields [9]. Underwater autonomous gliders for ocean sampling cannot find usual geometrical models from measurements of environmental variables such as pH, salinity, and temperature [10]. 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 paper, 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]–[14]. Random fields such as Gaussian processes and Gaussian Markov random fields (GMRFs) [15], [16] 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., [17]–[19].

The recent research efforts that are closely related to our problem are summarized as follows. In [20], the authors formulated Gaussian process regression under uncertain localization. In [21], 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 [20], [21]. In [22], 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 [22]. In [9], [23], 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 [24] 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 [8], 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 [8], 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 paper, we formulate the simultaneous localization and spatial prediction (SLAP) problem, in order to simultaneously localize robotic sensors and predict a spa-

Mahdi Jadaliha and Do Huan are with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA. E-mail: jadaliha@msu.edu.

Jongeun Choi is with the Department of Mechanical Engineering and the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA. E-mail: jchoi@egr.msu.edu.

tial 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.

Standard notation will be used throughout the paper. Let $\mathbb{R}$ and $\mathbb{Z}_{>0}$ denote, respectively, the sets of real and positive integer numbers. The collection of n number of m dimensional vectors $\{q_i \in \mathbb{R}^m \mid i = 1, \cdots, n\}$ can be defined by $q := \mathrm{col}\,(q_1, \cdots, q_n) \in \mathbb{R}^{nm}$ using the notation $\mathrm{col}(\cdot)$. The operators of expectation and covariance matrix are denoted by $\mathbb{E}$ and Cov, respectively. A random vector x, which has a multivariate normal distribution of mean vector μ and covariance matrix Σ, is denoted by $x \sim \mathcal{N}(\mu, \Sigma)$. For given $G = \{c, d\}$ and $H = \{1, 2\}$, the multiplication between two sets is defined as $H \times G = \{(1, c), (1, d), (2, c), (2, d)\}$. Other notation will be explained in due course.

II. Sequential Baysian inference with a GMRF

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

A. 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: $z(q) \sim \mathcal{GP}(\mu, \Sigma)$, where μ is the mean vector, and $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance matrix. We discretize the compact domain $\mathcal{S}_c := [0\ x_{max}] \times [0\ y_{max}]$ into n spatial sites $\mathcal{S} := \{s^{[1]}, \cdots, s^{[n]}\} \subset \mathbb{R}^d$, where $n = hx_{max} \times hy_{max}$. 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 $\mathcal{S}$ is denoted by $z := (z^{[1]}, \cdots, z^{[n]})^T \in \mathbb{R}^n$, where $z^{[i]} := z(s^{[i]})$.

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

$$\pi(z) \propto \exp\left(-\frac{1}{2}\left(z-\mu\right)^T \Sigma^{-1}\left(z-\mu\right)\right). \tag{1}$$

The i, j-th element of Σ is defined as $\Sigma^{[ij]} = \mathrm{Cov}(z^{[i]}, z^{[j]})$. The prior distribution of z can be written by a precision matrix $Q = \Sigma^{-1}$, i.e., $z \sim \mathcal{N}(\mu, Q^{-1})$. This can be viewed as a discretized version of the Gaussian process (or a GMRF) with a precision matrix Q on $\mathcal{S}$. Note that Q of this GMRF is not sparse. However, a sparse version of Q, i.e., $\hat{Q}$ with local neighborhood that can represent the original Gaussian process can be found, for example, making $\hat{Q}$ close to Q in some norm [25]–[27]. This approximate GMRF will be computationally efficient due to the sparsity of $\hat{Q}$. In our simulation study, we will use a GMRF with a sparse precision matrix that represents a Gaussian process precisely as shown in [21], [28]. However, any parameterization of μ_θ and Q_θ, where θ is the hyperparameter vector, can be used.

B. 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 $\mathcal{S}$. Let n spatial sites in $\mathcal{S}$ be indexed by $\mathcal{I} := \{1, \cdots, n\}$, and $z := \mathrm{col}\left(z^{[1]}, \cdots, z^{[n]}\right) \in \mathbb{R}^n$ be the corresponding static values of the scalar field at n special sites. We denote all robots' locations at time t by $q_t = \mathrm{col}\left({q_t}^{[1]}, \cdots, {q_t}^{[N]}\right) \in \mathcal{S}^N$, the observations made by all robots at time t by $\tilde{z}_t = \mathrm{col}\left(\tilde{z}_t^{[1]}, \cdots, \tilde{z}_t^{[N]}\right) \in \mathbb{R}^N$, and the observed states of all robots at time t by $\tilde{q}_t = \mathrm{col}\left(\tilde{q}_t^{[1]}, \cdots, \tilde{q}_t^{[N]}\right)$. $\tilde{q}_t$ and $\tilde{z}_t$ are noisy observations of q_t and z, respectively. At time t, robot j takes a noise corrupted measurement at its current location ${q_t}^{[j]} = s^{[i]} \in \mathcal{S}, \forall j \in \mathcal{J}, i \in \mathcal{I}$, viz.,

$$\tilde{z}_t^{[j]} = z^{[i]} + \epsilon_t^{[j]}, \tag{2}$$

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 $\epsilon_t := \mathrm{col}\left(\epsilon_t^{[1]}, \cdots, \epsilon_t^{[N]}\right) \in \mathbb{R}^N$.

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

$$\tilde{q}_t^{[j]} = q_t^{[j]} + e_t^{[j]}, \tag{3}$$

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 $e_t := \mathrm{col}\left(e_t^{[1]}, \cdots, e_t^{[N]}\right) \in \mathbb{R}^{d \times N}$.

Our models can be represented in the concise collective notation.

$$\begin{aligned} \tilde{z}_t &= H_{q_t} z + \epsilon_t, \\ \tilde{q}_t &= L_t q_t + e_t, \end{aligned} \tag{4}$$

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

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

C. 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., $\mathcal{S} \in \mathbb{R}^2$. In this case, an equation of motion for robot i [29] may be given by

$$\begin{bmatrix} \dot{q}_t^{[1,i]} \\ \dot{q}_t^{[2,i]} \end{bmatrix} = \begin{bmatrix} u_t^{[i]} \cos \psi_t^{[i]} \\ u_t^{[i]} \sin \psi_t^{[i]} \end{bmatrix} + \beta_t^{[i]}, \tag{5}$$

where $\{q_t^{[1,i]}, q_t^{[2,i]}\}$, $\{\psi_t^{[i]}\}$, $\{u_t^{[i]}\}$, 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.

$$q_{t+1} = q_t + F_t u_t + w_t, \tag{6}$$

where u_t is a known control input and w_t is an i.i.d. white noise realized by a known normal distribution $\mathcal{N}(0, \Sigma_{w_t})$. Assuming that $\{\psi_t^{[i]} | \forall i \in \mathcal{J}\}$ in (5) can be measured precisly, $F_t \in \mathbb{R}^{2N \times N}$ in (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 Δt 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 $\tilde{q}_{1:t} := \mathrm{col}(\tilde{q}_1, \cdots, \tilde{q}_t) \in \mathcal{S}_c^{Nt}$, $\tilde{z}_{1:t} := \mathrm{col}(\tilde{z}_1, \cdots, \tilde{z}_t) \in \mathbb{R}^{Nt}$, and $u_{1:t} := \mathrm{col}(u_1, \cdots, u_t) \in \mathbb{R}^{Nt}$.

D. 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 z is generated by a GMRF model which is given by $z \sim \mathcal{N}(\mu_\theta, Q_\theta^{-1})$, where μ_θ and Q_θ are given functions of a hyperparameter vector θ.

A.2 The noisy measurements $\{\tilde{z}_t\}$ and the noisy sampling positions $\{\tilde{q}_t\}$, as in (4), are collected by robotic sensors in time $t = 1, 2, \cdots$.

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

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

A.5 The prior distribution of the sampling positions in time t, $\pi(q_t)$, is discrete with a support $\Omega(t) = \{q_t^{(k)} | k \in \mathcal{L}(t)\}$, which is given at time t. Here, $\mathcal{L}(t) = \{1, \cdots, \gamma(t)\}$ denotes the index in the support and $\gamma(t)$ is the number of the probable possibilities for q_t.

Problem 2.1: Consider the assumptions A.1-A.5. Our problem is to simultaneously find the predictive distributions, means, and variances for both z and q conditional on $\mathcal{D}_t := \{\tilde{z}_{1:t}, \tilde{q}_{1:t}\}$.

The solution to Problem 2.1 is derived as follows. The distribution of the GMRF is given by $\pi(z|\theta, \mathcal{D}_{t-1}) = \mathcal{N}\left(\mu_{z|\theta,\mathcal{D}_{t-1}}, \Sigma_{z|\theta,\mathcal{D}_{t-1}}\right)$. Recall that the evolution of q_t is given by (6) and the input u_t is a known deterministic vector at time t. Therefore, $\pi(q_t|\mathcal{D}_{t-1})$ can be updated by the Gaussian approximation of $\pi(q_{t-1|\mathcal{D}_{t-1}})$.

$$\begin{aligned} &\pi(q_t|\mathcal{D}_{t-1}) \approx \\ &\quad \mathcal{N}\left(\mu_{q_{t-1}|\mathcal{D}_{t-1}} + F_{t-1}u_{t-1}, \Sigma_{q_{t-1}|\mathcal{D}_{t-1}} + \Sigma_{w_{t-1}}\right). \end{aligned} \tag{7}$$

Similarly, $\pi(\tilde{z}_t|\theta, \mathcal{D}_{t-1}, q_t)$ is updated by the Gaussian approximation of $\pi(z|\theta, \mathcal{D}_{t-1})$ as follows.

$$\begin{aligned} &\pi(\tilde{z}_t|\theta, \mathcal{D}_{t-1}, q_t) \approx \\ &\quad \mathcal{N}\left(H_{q_t}\mu_{z|\theta,\mathcal{D}_{t-1}}, \Sigma_{\epsilon_t} + H_{q_t}\Sigma_{z|\theta,\mathcal{D}_{t-1}}H_{q_t}^T\right). \end{aligned} \tag{8}$$

Remark 2.2: For the sake of reducing memory usage and complexity, the distribution of $q_t|\mathcal{D}_{t-1}$ and $\tilde{z}_t|\theta, \mathcal{D}_{t-1}, q_t$ are approximated by normal distributions.

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

$$\begin{aligned} &\pi(z, q_t, \theta|\mathcal{D}_{t-1}) = \\ &\quad \pi(z|\theta, q_t, \mathcal{D}_{t-1})\,\pi(\theta|q_t, \mathcal{D}_{t-1})\,\pi(q_t|\mathcal{D}_{t-1}). \end{aligned} \tag{9}$$

The observation model is given by (4), thus the probabilities of the observed data are $\pi(\tilde{z}_t|z, q_t) = \mathcal{N}(H_{q_t}z, \Sigma_{\varepsilon_t})$ and $\pi(\tilde{q}_t|q_t) = \mathcal{N}(L_t q_t, \Sigma_{e_t})$. The measured random variables have the following conditional joint distribution,

$$\pi(\tilde{z}_t, \tilde{q}_t|z, q_t, \theta, \mathcal{D}_{t-1}) = \pi(\tilde{z}_t|z, \theta, q_t, \mathcal{D}_{t-1})\,\pi(\tilde{q}_t|q_t). \tag{10}$$

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

$$\pi(z, q_t, \theta|\mathcal{D}_t) = \frac{\pi(\tilde{z}_t, \tilde{q}_t|z, q_t, \theta, \mathcal{D}_{t-1})\,\pi(z, q_t, \theta|\mathcal{D}_{t-1})}{\pi(\tilde{z}_t, \tilde{q}_t|D_{t-1})}. \tag{11}$$

In addition, $\pi(q_t, \theta|\mathcal{D}_t) = \int \pi(z, q_t, \theta|\mathcal{D}_t)\,dz$ is given as follows.

$$\begin{aligned} &\int \pi(z, q_t, \theta|\mathcal{D}_t)\,dz = \\ &\quad \frac{\pi(\tilde{q}_t|q_t)\,\pi(\theta|q_t, \mathcal{D}_{t-1})\,\pi(q_t|\mathcal{D}_{t-1})}{\pi(\tilde{z}_t, \tilde{q}_t|D_{t-1})} \times \\ &\quad \int \pi(z|\theta, q_t, \mathcal{D}_{t-1})\,\pi(\tilde{z}_t|z, \theta, q_t, \mathcal{D}_{t-1})\,dz, \end{aligned} \tag{12}$$

where $\int \pi(z|\theta, q_t, \mathcal{D}_{t-1})\,\pi(\tilde{z}_t|z, \theta, q_t, \mathcal{D}_{t-1})\,dz = \pi(\tilde{z}_t|q_t, \theta, \mathcal{D}_{t-1})$, and $\pi(\tilde{z}_t|q_t, \theta, \mathcal{D}_{t-1})$ is given by (8).

Remark 2.3: From the Bayes' rule, $\pi(\theta|q_t, \mathcal{D}_{t-1})$ is given by $\frac{\pi(\theta, q_t|\mathcal{D}_{t-1})}{\pi(q_t|\mathcal{D}_{t-1})}$. We can compute $\pi(\theta, q_t|\mathcal{D}_{t-1})$ for all the possible combinations of q_t in the previous iteration using (12). However, for the sake of reducing the computational cost, we approximate $\pi(\theta|q_t, \mathcal{D}_{t-1})$ by $\pi(\theta|\mathcal{D}_{t-1})$. Therefore, we have

$$\begin{aligned} &\pi(q_t, \theta|\mathcal{D}_t) \approx \\ &\quad \frac{\pi(\tilde{q}_t|q_t)\,\pi(\theta|D_{t-1})\,\pi(q_t|D_{t-1})\pi(\tilde{z}_t|q_t, \theta, \mathcal{D}_{t-1})}{\pi(\tilde{z}_t, \tilde{q}_t|D_{t-1})}, \end{aligned}$$

Marginalizing out uncertainties on the possible q_t and θ, we obtain the following.

$$\begin{aligned} \pi(z, \theta|\mathcal{D}_t) &= \sum_{q_t \in \Omega(t)} \pi(z, q_t, \theta|\mathcal{D}_t), \\ \pi(z|\mathcal{D}_t) &= \sum_{\theta \in \Theta} \pi(z, \theta|\mathcal{D}_t). \end{aligned} \tag{13}$$

Our estimation of q_t and θ can be corrected using measured data up to time t as follows.

$$\begin{aligned} \pi\left(q_t|\mathcal{D}_t\right) &= \sum_{\theta\in\Theta}\pi\left(q_t,\theta|\mathcal{D}_t\right), \\ \pi\left(\theta|\mathcal{D}_t\right) &= \sum_{q_t\in\Omega(t)}\pi\left(q_t,\theta|\mathcal{D}_t\right). \end{aligned} \tag{14}$$

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

$$\begin{aligned} \pi\left(z|\theta,\mathcal{D}_t\right) &= \frac{\pi\left(z,\theta|\mathcal{D}_t\right)}{\pi\left(\theta|\mathcal{D}_t\right)}, \\ \mu_{z|\theta,\mathcal{D}_t} &= \frac{1}{\pi\left(\theta|\mathcal{D}_t\right)}\sum_{q_t\in\Omega(t)}\mu_{z|q_t,\theta,\mathcal{D}_t}\pi\left(q_t,\theta|\mathcal{D}_t\right). \end{aligned} \tag{15}$$

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

$$\begin{aligned} \mathbb{E}\left(\Sigma_{z|q_t,\theta,\mathcal{D}_t}\right) &= \sum_{q_t\in\Omega(t)}\Sigma_{z|q_t,\theta,\mathcal{D}_t}\pi\left(q_t|\theta,\mathcal{D}_t\right), \\ \mathrm{Cov}\left(\mu_{z|q_t,\theta,\mathcal{D}_t}\right) &= \sum_{q_t\in\Omega(t)}\left(\mu_{z|q_t,\theta,\mathcal{D}_t}-\mu_{z|\theta,\mathcal{D}_t}\right)\times \\ &\quad \left(\mu_{z|q_t,\theta,\mathcal{D}_t}-\mu_{z|\theta,\mathcal{D}_t}\right)^T\pi\left(q_t|\theta,\mathcal{D}_t\right), \end{aligned} \tag{16}$$

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

$$\begin{aligned} &\mu_{z|q_t,\theta,\mathcal{D}_t} = \mu_{z|\theta,\mathcal{D}_{t-1}} \\ &\quad + \Sigma_{z|\theta,\mathcal{D}_{t-1}}H_{q_t}^T\Sigma_{\tilde{z}_t|\theta,\mathcal{D}_{t-1},q_t}^{-1}(\tilde{z}_t-\mu_{\tilde{z}_t|\theta,\mathcal{D}_{t-1},q_t}), \\ &\Sigma_{z|q_t,\theta,\mathcal{D}_t} = \Sigma_{z|\theta,\mathcal{D}_{t-1}} \\ &\quad - \Sigma_{z|\theta,\mathcal{D}_{t-1}}H_{q_t}^T\Sigma_{\tilde{z}_t|\theta,\mathcal{D}_{t-1},q_t}^{-1}H_{q_t}\Sigma_{z|\theta,\mathcal{D}_{t-1}}. \end{aligned} \tag{17}$$

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

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

III. 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 $\mathcal{S}$. The spatial sites in $\mathcal{S}$ consist of 31×31 grid points, i.e., $n = 961$, uniformly distributed over the surveillance region $\mathcal{S}_c := [-15,15]\times[-15,15]$. The evolution of the location of the robot can be more detailed as follows.

$$\begin{aligned} q_{t+1} &= Q\left(q_t + F_t u_t + v_t\right) \\ &= q_t + F_t u_t + w_t, \end{aligned} \tag{18}$$

where $Q : \mathcal{S}_c \to \mathcal{S}$ is the nearest neighbor rule quantizer that takes an input and returns a projected value on $\mathcal{S}$. v_t is the process noise and w_t is the quantization error between the continuous and discretized states, i.e., $w_t = Q\left(q_t + F_t u_t + v_t\right) - (q_t + F_t u_t)$. As the cardinality of $\mathcal{S}$ increases, we have that $w_t \to v_t$. A special case of (18) is that $F_t u_t$ is controlled and w_t is chosen such that the next location q_{t+1} is on a grid point in $\mathcal{S}$. In this case, we have $v_t = w_t$.

In this illustrative example, we realize the spatial field developed in [21], 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 [21] 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 (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. 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 (3). The probable sampling positions that form support $\Omega(t)$, are selected within the confidence region of $\Pr(\|q_t^{[i]} - \tilde{q}_t^{[i]}\| \leq \sigma_e)$.

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

- *Case 1:* Figs. 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. 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. 1-(h) shows that squared error of this case is considerably higher than that of Case 1.
- *Case 3:* Figs. 1-(c), (f), and (i) show the results, by applying the proposed approach in Section II-D. 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 2.1.

The true positions of the robot for time $t \in \mathcal{T} := \{10,31,\cdots,30\}$ are shown in Fig. 2 by red diamonds and lines. The estimated sampling positions of the robot $\mathbb{E}(q_t|\mathcal{D}_t)$

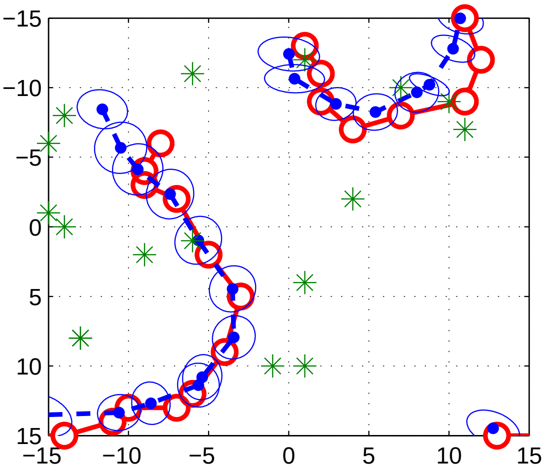

Fig. 2. The trajectories of true, predicted, and noisy sampling positions of the robot are shown by red diamonds, blue dots, and green stars for time $t \in \{10, 11, \cdots, 30\}$. The blue ellipsoids show the confidence regions of about 68% for the estimated sampling positions.

for $t \in \mathcal{T}$ are shown in blue dots with estimated confidence regions. Fig. 2 clearly shows that the proposed approach in this paper 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.

IV. The Body of the Article

A. Mathematics

asjcauth.cls makes the full functionality of $\mathcal{AMS}$TEX available. We encourage the use of the `align`, `gather` and `multline` environments for displayed mathematics.

B. Two wheel robot model

Call q_{ci}, θ_i are current position and orientation of the robot. v_{ci} and ω_i are linear and angular velocity.

$$\begin{aligned} \dot{q}_{ci} &= \begin{bmatrix} v_{ci}\cos(\theta_i) \\ v_{ci}\sin(\theta_i) \end{bmatrix}, \\ \dot{\theta}_i &= \omega_i \end{aligned} \tag{19}$$

With

$$\mathbf{\dot{q}_{ci}} = \begin{bmatrix} v_{cx} \\ v_{cy} \end{bmatrix} \tag{20}$$

Combining (1) and (2), we have:

$$\begin{bmatrix} v_{cx} \\ v_{cy} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & 0 \\ \sin(\theta_i) & 0 \end{bmatrix} \begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix}, \tag{21}$$

Where v_{cx} and v_{cy} are velocity of the center point of the robot in *x-axis* and *y-axis* of the *global coordinate*. Calling:

$$M(\theta) = \begin{bmatrix} \cos(\theta_i) & 0 \\ \sin(\theta_i) & 0 \end{bmatrix} \tag{22}$$

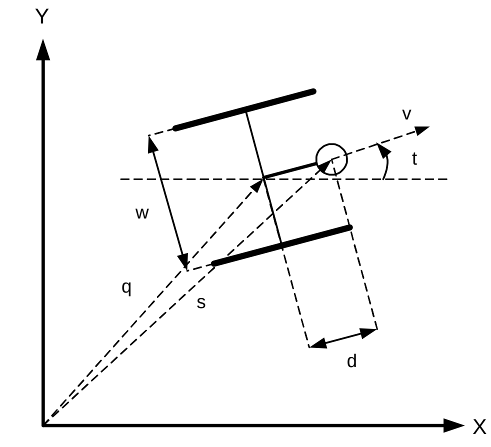

Fig. 3. Robot's Kinematics

In the discrete domain, we have:

$$\frac{q_i(t+1) - q_i(t)}{\Delta t} = M(\theta) \begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix} \tag{23}$$

Thus,

$$q_i(t+1) = q_i(t) + \Delta t M(\theta) \begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix}, \tag{24}$$

$$\theta_i(t+1) = \theta_i(t) + \Delta t \omega_i(t) \tag{25}$$

On other hand, call v_l and v_r are velocity of the left and right wheels of the robot, correspondingly, we have:

$$\begin{aligned} v_{ci} &= \frac{v_l + v_r}{2} \\ \omega_{ci} &= \frac{v_l - v_r}{d_w} \end{aligned} \tag{26}$$

With d_w is the distance between two wheel.So:

$$\begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{d_w} & \frac{-1}{d_w} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \tag{27}$$

Shown in Figure 2 is the diagram for the two wheel mobile robot.

C. Experiment setup

1) Mobile robot: The mobile robot is depicted in Figure 3. The Arduino Mega board is used as the micro-controller, two wr703n router are used, one for streaming the video recorded from the web-cam equipped with paranormal lens, the other for receiving command remotely via the Internet. The overall control scheme is elaborated in Figure 4. There are three data collecting processes:

- The command sent to the Arduino board from the user is harvested at sampling time set to 50 milliseconds. However, due to the poor performance of the Arduino board, the actual sampling time fluctuates from 59 to 62 milliseconds. Thus,collapsing time between two samplings are recorded to be used in the simulation part.
- The scene recorded by the paranormal web-cam is streamed on-line via the router, which can be accessible by the IP address of the router. The control program

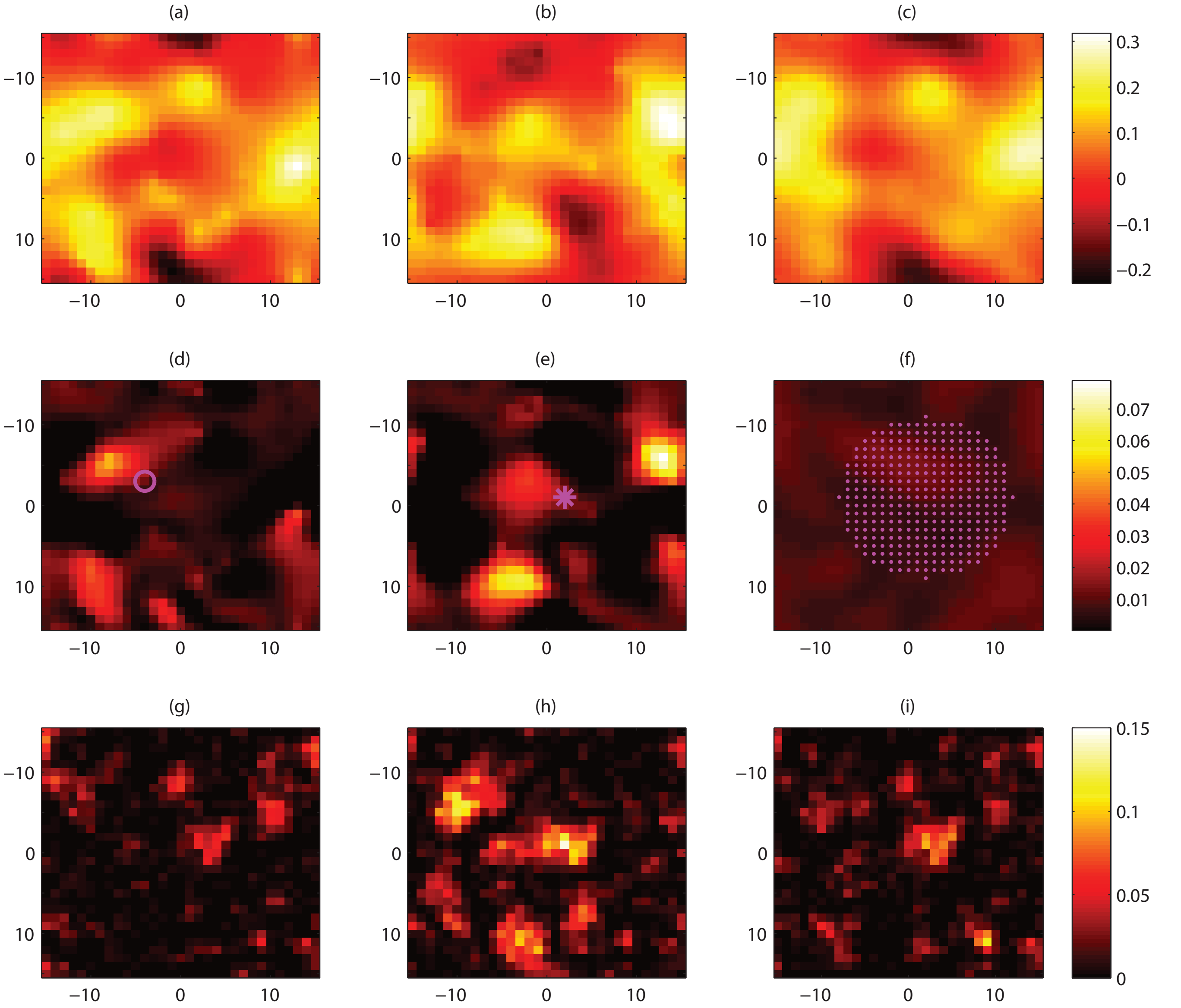

Fig. 1. The prediction results of Cases 1, 2, and 3 at time $t = 100$ are shown in the first, second, third columns, respectively. The first, second, and third rows correspond to the prediction, prediction error variance, and squared empirical error fields between the prediction and true fields. True, 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.

running in the control PC records the stream with 5 frame per second, e.g. 5 Hz.

- The location of the mobile robot is tracked by the ceiling camera, by which it is updated with the same sampling rate as the streaming video.

2) Experiment environment: The testing environment is depicted in Figure 5. Robot is controlled in opened-loop manner by user, while the paranormal recording the surrounding scene.Two PCs are used in the experiment, one for sending command signals to the robot, the other for processing the images from the ceiling camera for tracking robot's trajectory, and recording the scene streamed from the paranormal camera installed on the robot.

D. mat-lab code explanation

The command saved from the Arduino board is stored in *huan* file. The real robot's trajectory *in image frame coordinate* is exported in *output.txt* file. The scene recorded from the paranormal camera is saved in *record12115h08.avi* file.

The file *robotTrackSimulation.m* simulates the robot model which is described in 5.1. The simulated result is converted to the coordinate in picture's frame by the camera matrix stored in the file *calibmatrix.mat*. The trajectory then is processed through a second calibration by translation and scaling matrices to minimize the difference between the simulated and the real trajectory. The model is confirmed

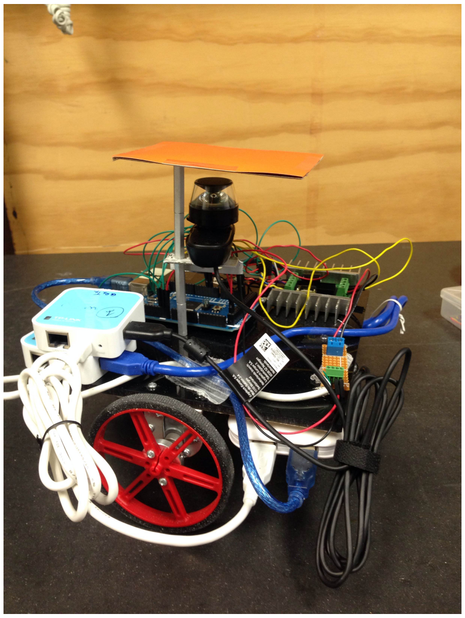

Fig. 4. Mobile Robot

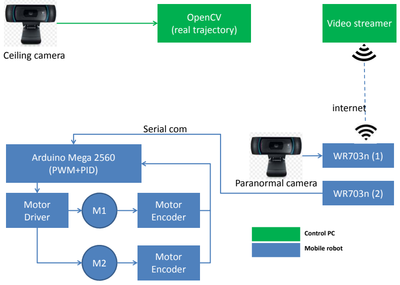

Fig. 5. Overall Control Scheme

Fig. 6. Testing Environment

when the distance is sufficiently small.

The frames are queried from the video in the file *frameQuery.m*. The experiment is executed in 112 seconds, e.g. sum of the sampling time from the command file and the length of the video is exactly matched. The command is sampled with 60 milliseconds sampling time, so there are 1852 sample points, which in turn are sub-sampled by factor of 1:4. Thus, there are **463** sub-sampling points. The video is recorded with 5 fps, so there are **558** frames for the video in total. The frames are numbered from 1 to 558. The sub-sampled command input is stored in the **commandSS** variable which is 463x3 matrix. The first two columns are the command input of linear velocity and angular velocity for the robot, correspondingly, the final column is exact time when a command was sent, in the time-line from 0 to 112.5 second. The variable **outPut** contains the real coordinate of the robot, sampled with frequency of 5Hz.

The frames and the sub-sampled command are matched in following manner:

The time (in second) for each sample point in 463 sample points of the command is calculated in the time-line from 0 to 111 second, then it is projected to the frames set by following equation:

$$framenumber = time \times \frac{558}{111} \tag{28}$$

The 463 sample frames are save in the *frames* folder.

V. Conclusion

In this paper, 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 [21], 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.

VI. Acknowledgement

This work has been supported by the National Science Foundation through CAREER Award CMMI-0846547. This support is gratefully acknowledged.

References

[1] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

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

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

[4] J. Guivant and E. Nebot, “Optimization of the simultaneous localization and map-building algorithm for real-time implementation,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[5] 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*, vol. 18, 2003, pp. 1151–1156.

[6] K. Murphy, “Bayesian map learning in dynamic environments,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, pp. 1015–1021, 1999.

[7] 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*. IEEE, 2005, pp. 2432–2437.

[8] 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*, 2007, pp. 2480–2485.

[9] 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*, September 2010, pp. 14–19.

[10] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis, “Collective motion, sensor networks, and ocean sampling,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, January 2007.

[11] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, “Decentralized environmental modeling by mobile sensor networks,” *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, June 2008.

[12] J. Choi, S. Oh, and R. Horowitz, “Distributed learning and cooperative control for multi-agent systems,” *Automatica*, vol. 45, no. 12, pp. 2802–2814, December 2009.

[13] J. Cortés, “Distributed Kriged Kalman filter for spatial estimation,” *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816–2827, December 2009.

[14] 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*, vol. 57, no. 8, pp. 2078–2084, 2012.

[15] N. Cressie, “Kriging nonstationary data,” *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 625–634, September 1986.

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

[17] 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*, vol. 9, pp. 235–284, June 2008.

[18] Y. Xu, J. Choi, and S. Oh, “Mobile sensor network navigation using Gaussian processes with truncated observations,” *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1118 – 1131, December 2011.

[19] R. Graham and J. Cortés, “Cooperative adaptive sampling of random fields with partially known covariance,” *International Journal of Robust and Nonlinear Control*, vol. 22, no. 5, pp. 504–534, March 2012.

[20] 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, pp. 1394–1399.

[21] ——, “Efficient Spatial Prediction Using Gaussian Markov Random Fields Under Uncertain Localization,” in *Proceeding of the ASME Dynamic Systems and Control Conference*, October 2012.

[22] A. Brooks, A. Makarenko, and B. Upcroft, “Gaussian process models for indoor and outdoor sensor-centric robot localization,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1341–1351, December 2008.

[23] 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*, September 2010, pp. 7–13.

[24] S. O’Callaghan, F. Ramos, and H. Durrant-Whyte, “Contextual occupancy maps using Gaussian processes,” in *Proceeding of the IEEE International Conference on Robotics and Automation*, May 2009, pp. 1054–1060.

[25] H. Rue and H. Tjelmeland, “Fitting Gaussian Markov random fields to Gaussian fields,” *Scandinavian Journal of Statistics*, vol. 29, no. 1, pp. 31–49, March 2002.

[26] N. Cressie and N. Verzelen, “Conditional-mean least-squares fitting of Gaussian Markov random fields to Gaussian fields,” *Computational Statistics & Data Analysis*, vol. 52, no. 5, pp. 2794–2807, January 2008.

[27] L. Hartman and O. Hössjer, “Fast kriging of large data sets with Gaussian Markov random fields,” *Computational Statistics & Data Analysis*, vol. 52, no. 5, pp. 2331–2349, January 2008.

[28] 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*, vol. 73, no. 4, pp. 423–498, September 2011.

[29] W. Ren, “Consensus strategies for cooperative control of vehicle formations,” *Control Theory & Applications, IET*, vol. 1, no. 2, pp. 505–512, 2007.