# GP-Localize: Persistent Mobile Robot Localization using Online Sparse Gaussian Process Observation Model

**Nuo Xu**[†] and **Kian Hsiang Low**[†] and **Jie Chen**[§] and **Keng Kiat Lim**[†] and **Etkin Barış Özgül**[†]

Department of Computer Science, National University of Singapore, Republic of Singapore[†]
Singapore-MIT Alliance for Research and Technology, Republic of Singapore[§]
{xunuo, lowkh, kengkiat, ebozgul}@comp.nus.edu.sg[†], chenjie@smart.mit.edu[§]

## Abstract

Central to robot exploration and mapping is the task of persistent localization in environmental fields characterized by spatially correlated measurements. This paper presents a *Gaussian process localization* (GP-Localize) algorithm that, in contrast to existing works, can exploit the spatially correlated field measurements taken during a robot's exploration (instead of relying on prior training data) for efficiently and scalably learning the GP observation model online through our proposed novel online sparse GP. As a result, GP-Localize is capable of achieving *constant* time and memory (i.e., independent of the size of the data) per filtering step, which demonstrates the practical feasibility of using GPs for persistent robot localization and autonomy. Empirical evaluation via simulated experiments with real-world datasets and a real robot experiment shows that GP-Localize outperforms existing GP localization algorithms.

## 1 Introduction

Recent research in robot exploration and mapping has focused on developing adaptive sampling and active sensing algorithms (Cao, Low, and Dolan 2013; Chen, Low, and Tan 2013; Chen et al. 2012; Hoang et al. 2014; Low, Dolan, and Khosla 2008; 2009; 2011; Low et al. 2007; 2012; Ouyang et al. 2014) to gather the most informative data/observations for modeling and predicting spatially varying environmental fields that are characterized by *continuous-valued*, *spatially correlated* measurements. Application domains (e.g., environmental sensing and monitoring) requiring such algorithms often contain multiple fields of interest: (a) Autonomous underwater and surface vehicles are tasked to sample ocean and freshwater phenomena including temperature, salinity, and oxygen concentration fields (Dolan et al. 2009; Podnar et al. 2010), (b) indoor environments are spanned by temperature, light, and carbon dioxide concentration fields that affect the occupants' comfort and satisfaction towards the environmental quality across different areas, and (c) WiFi access points/hotspots situated at neighboring locations produce different but overlapping wireless signal strength fields over the same environment. These algorithms operate with an assumption that the locations of every robot and its gathered observations are known and provided by its onboard sensors such as the widely-used GPS

device. However, GPS signals may be noisy (e.g., due to urban canyon effect between tall buildings) or unavailable (e.g., in underwater or indoor environments). So, it is desirable to alternatively consider exploiting the spatially correlated measurements taken by each robot for localizing itself within the environmental fields during its exploration; this will significantly extend the range of environments and application domains in which a robot can localize itself.

To achieve this, our robotics community will usually make use of a probabilistic state estimation framework known as the *Bayes filter*: It repeatedly updates the belief of a robot's location/state by assimilating the field measurements taken during the robot's exploration through its *observation model*. To preserve time efficiency, the Bayes filter imposes a Markov property on the observation model: Given the robot's current location, its current measurement is conditionally independent of the past measurements. Such a Markov property is severely violated by the spatial correlation structure of the environmental fields, thus strongly degrading the robot's localization performance. To resolve this issue, the works of Ko and Fox (2009a; 2009b) have integrated a rich class of Bayesian nonparametric models called the *Gaussian process* (GP) into the Bayes filter, which allows the spatial correlation structure between measurements to be formally characterized (i.e., by modeling each field as a GP) and the observation model to be represented by fully probabilistic predictive distributions (i.e., one per field/GP) with formal measures of the uncertainty of the predictions.

Unfortunately, such expressive power of a GP comes at a high computational cost, which hinders its practical use in the Bayes filter for persistent robot localization: It incurs cubic time and quadratic memory in the size of the data/observations. Existing works (Brooks, Makarenko, and Upcroft 2008; Ferris, Hähnel, and Fox 2006; Ferris, Fox, and Lawrence 2007; Ko and Fox 2009a; 2009b) have sidestepped this computational difficulty by assuming the availability of data/observations *prior* to exploration and localization for training the GP observation model offline; some (Brooks, Makarenko, and Upcroft 2008; Ferris, Hähnel, and Fox 2006; Ko and Fox 2009a) have assumed these given prior measurements to be labeled with known locations while others (Ferris, Fox, and Lawrence 2007; Ko and Fox 2009b) have inferred their location labels. The Markov assumption on the observation model can then be

"relaxed" to conditional independence between the robot's current measurement and past measurements (i.e., taken during its exploration) given its current location and the trained GPs using prior data/observations, thus improving the efficiency at each filtering step during its exploration to quadratic time in the size of the prior training data. Any measurement taken during the robot's actual exploration and localization is thus not used to train the GP observation model. Such a "relaxed" Markov assumption may hold in certain static environments. However, it becomes highly restrictive and is easily violated in general, practical environmental settings where, for example, (a) limited sampling budget (i.e., in terms of energy consumption, mission time, etc.) forbids the collection of prior training data or only permits extremely sparse prior data to be gathered relative to a large environment, thus resulting in an inaccurately trained GP observation model, (b) environmental changes invalidate the prior training data, and (c) the robot's actual exploration path is spatially distant from the prior observations, hence making the trained GP observation model uninformative to its localization. All these practical considerations motivate us to tackle a fundamental research question: Without prior training data, how can GPs be restructured to be used by a Bayes filter for *persistent* robot localization in environmental fields characterized by spatially correlated measurements?

This paper presents a *Gaussian process localization* (GP-Localize) algorithm that, in contrast to existing works mentioned above, can exploit the spatially correlated field measurements taken during a robot's exploration (instead of relying on prior training data) for efficiently and scalably learning the GP observation model online through our proposed novel online sparse GP (Section 3). As a result, GP-Localize is capable of achieving *constant* time and memory (i.e., independent of the size of the data/observations) per filtering step, which we believe is an important first step towards demonstrating the practical feasibility of employing GPs for persistent robot localization and autonomy. We empirically demonstrate through simulated experiments with three real-world datasets as well as a real robot experiment that GP-Localize outperforms existing GP localization algorithms (Section 4).

## 2 Background

### 2.1 Modeling Environmental Field with GP

The Gaussian process (GP) can be used to model an environmental field as follows[1]: The environmental field is defined to vary as a realization of a GP. Let $\mathcal{X}$ be a set of locations representing the domain of the environmental field such that each location $x \in \mathcal{X}$ is associated with a realized (random) field measurement $z_x(Z_x)$ if $x$ is observed (unobserved). Let $\{Z_x\}_{x \in \mathcal{X}}$ denote a GP, that is, every finite subset of $\{Z_x\}_{x \in \mathcal{X}}$ has a multivariate Gaussian distribution (Rasmussen and Williams 2006). The GP is fully specified by its *prior* mean $\mu_x \triangleq \mathbb{E}[Z_x]$ and covariance $\sigma_{xx'} \triangleq \mathrm{cov}[Z_x, Z_{x'}]$ for all $x, x' \in \mathcal{X}$, the latter of which characterizes the spatial correlation structure of the field

---

[1]To simplify exposition, we only describe the GP for a single field; for multiple fields, we assume independence between them to ease computations.

and can be defined using a covariance function. A common choice is the squared exponential covariance function $\sigma_{xx'} \triangleq \sigma_s^2 \exp\{-0.5(x-x')^\top M^{-2}(x-x') + \sigma_n^2 \delta_{xx'}\}$ where $\sigma_s^2$ and $\sigma_n^2$ are, respectively, the signal and noise variance controlling the intensity and the noise of the measurements, $M$ is a diagonal matrix with length-scale components $\ell_1$ and $\ell_2$ controlling, respectively, the degree of spatial correlation or "similarity" between measurements in the horizontal and vertical directions of the field, and $\delta_{xx'}$ is a Kronecker delta of value 1 if $x = x'$, and 0 otherwise.

A chief advantage of using the full GP to model the environmental field is its capability of performing probabilistic regression: Supposing a robot has visited and observed a set $\mathcal{D}$ of locations and taken a column vector $z_\mathcal{D}$ of corresponding realized measurements, the full GP can exploit these observations to predict the measurement at any unobserved location $x \in \mathcal{X} \setminus \mathcal{D}$ as well as provide its corresponding predictive uncertainty using a Gaussian predictive distribution $p(z_x | x, \mathcal{D}, z_\mathcal{D}) = \mathcal{N}(\mu_{x|\mathcal{D}}, \sigma_{xx|\mathcal{D}})$ with the following *posterior* mean and variance, respectively:

$$\mu_{x|\mathcal{D}} \triangleq \mu_x + \Sigma_{x\mathcal{D}} \Sigma_{\mathcal{D}\mathcal{D}}^{-1} (z_\mathcal{D} - \mu_\mathcal{D}) \tag{1}$$

$$\sigma_{xx|\mathcal{D}} \triangleq \sigma_{xx} - \Sigma_{x\mathcal{D}} \Sigma_{\mathcal{D}\mathcal{D}}^{-1} \Sigma_{\mathcal{D}x} \tag{2}$$

where $\mu_\mathcal{D}$ is a column vector with mean components $\mu_{x'}$ for all $x' \in \mathcal{D}$, $\Sigma_{x\mathcal{D}}$ is a row vector with covariance components $\sigma_{xx'}$ for all $x' \in \mathcal{D}$, $\Sigma_{\mathcal{D}x}$ is the transpose of $\Sigma_{x\mathcal{D}}$, and $\Sigma_{\mathcal{D}\mathcal{D}}$ is a matrix with components $\sigma_{x'x''}$ for all $x', x'' \in \mathcal{D}$.

### 2.2 Sparse Gaussian Process Approximation

The key limitation hindering the practical use of the full GP in the Bayes filter for persistent robot localization is its poor scalability in the size $|\mathcal{D}|$ of the data/observations: Computing the Gaussian predictive distribution (i.e., (1) and (2)) requires inverting the covariance matrix $\Sigma_{\mathcal{D}\mathcal{D}}$, which incurs $\mathcal{O}(|\mathcal{D}|^3)$ time and $\mathcal{O}(|\mathcal{D}|^2)$ memory. To improve its scalability, GP approximation methods (Chen et al. 2012; 2013; Quiñonero-Candela and Rasmussen 2005) have been proposed, two of which will be described below.

The simple sparse *subset of data* (SoD) approximation method uses only a subset $\mathcal{S}$ of the set $\mathcal{D}$ of locations (i.e., $\mathcal{S} \subset \mathcal{D}$) observed and the realized measurements $z_\mathcal{S}$ taken by the robot to produce a Gaussian predictive distribution of the measurement at any unobserved location $x \in \mathcal{X} \setminus \mathcal{D}$ with the following posterior mean and variance, which are similar to that of full GP (i.e., by replacing $\mathcal{D}$ in (1) and (2) with $\mathcal{S}$):

$$\mu_{x|\mathcal{S}} = \mu_x + \Sigma_{x\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{S}}^{-1} (z_\mathcal{S} - \mu_\mathcal{S}) \tag{3}$$

$$\sigma_{xx|\mathcal{S}} = \sigma_{xx} - \Sigma_{x\mathcal{S}} \Sigma_{\mathcal{S}\mathcal{S}}^{-1} \Sigma_{\mathcal{S}x} . \tag{4}$$

The covariance matrix $\Sigma_{\mathcal{S}\mathcal{S}}$ is inverted using $\mathcal{O}(|\mathcal{S}|^3)$ time and $\mathcal{O}(|\mathcal{S}|^2)$ memory, which are independent of $|\mathcal{D}|$. The main criticism of SoD is that it does not exploit all the data for computing the Gaussian predictive distribution, thus yielding an unrealistic overestimate (4) of the predictive uncertainty (even with fairly redundant data and informative subset $\mathcal{S}$) (Quiñonero-Candela and Rasmussen 2005) and in turn an inaccurately trained observation model.

The sparse *partially independent training conditional* (PITC) approximation method is the most general form of a class of reduced-rank covariance matrix approximation

methods reported in (Quiñonero-Candela and Rasmussen 2005) exploiting the notion of a support set $\mathcal{S} \subset \mathcal{X}$. Unlike SoD, PITC can utilize all data (i.e., $\mathcal{D}$ and $z_{\mathcal{D}}$) to derive a Gaussian predictive distribution of the measurement at any $x \in \mathcal{X} \setminus \mathcal{D}$ with the following posterior mean and variance:

$$\mu_{x|\mathcal{D}}^{\text{PITC}} \triangleq \mu_x + \Gamma_{x\mathcal{D}}(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda)^{-1}(z_{\mathcal{D}} - \mu_{\mathcal{D}}) \quad (5)$$

$$\sigma_{xx|\mathcal{D}}^{\text{PITC}} \triangleq \sigma_{xx} - \Gamma_{x\mathcal{D}}(\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda)^{-1}\Gamma_{\mathcal{D}x} \quad (6)$$

where $\Gamma_{\mathcal{A}\mathcal{A}'} = \Sigma_{\mathcal{A}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{A}'}$ for all $\mathcal{A}, \mathcal{A}' \subset \mathcal{X}$ and $\Lambda$ is a block-diagonal matrix constructed from the $N$ diagonal blocks of $\Sigma_{\mathcal{D}\mathcal{D}|\mathcal{S}}$, each of which is a matrix $\Sigma_{\mathcal{D}_n\mathcal{D}_n|\mathcal{S}}$ for $n = 1, \cdots, N$ where $\mathcal{D} = \bigcup_{n=1}^{N} \mathcal{D}_n$. Also, unlike SoD, the support set $\mathcal{S}$ does not have to be observed. The covariance matrix $\Sigma_{\mathcal{D}\mathcal{D}}$ in (1) and (2) is approximated by a reduced-rank matrix $\Gamma_{\mathcal{D}\mathcal{D}}$ summed with the resulting sparsified residual matrix $\Lambda$ in (5) and (6). So, computing either $\mu_{x|\mathcal{D}}^{\text{PITC}}$ (5) or $\sigma_{xx|\mathcal{D}}^{\text{PITC}}$ (6), which requires inverting the approximated covariance matrix $\Gamma_{\mathcal{D}\mathcal{D}} + \Lambda$, incurs $\mathcal{O}(|\mathcal{D}|(|\mathcal{S}|^2 + (|\mathcal{D}|/N)^2))$ time and $\mathcal{O}(|\mathcal{S}|^2 + (|\mathcal{D}|/N)^2)$ memory. The sparse *fully independent training conditional* (FITC) approximation method is a special case of PITC where $\Lambda$ is a diagonal matrix constructed from $\sigma_{x'x'|\mathcal{S}}$ for all $x' \in \mathcal{D}$ (i.e., $N = |\mathcal{D}|$). FITC is previously employed by Ko and Fox (2009a) to speed up the learning of observation model with prior training data. But, the time incurred by PITC or FITC grows with increasing size of data. So, it is computationally impractical to use them directly to repeatedly train the observation model at each filtering step for persistent localization.

## 2.3 Bayes Filters

A Bayes filter is a probabilistic state estimation framework that repeatedly updates the belief of a robot's location/state by conditioning on its control actions performed and field measurements taken so far. Formally, let the robot's control action performed, its location visited and observed, and the corresponding realized field measurement taken at time/filtering step $t$ be denoted by $u_t$, $x_t$, and $z_t$[2], respectively. To estimate the robot's location, a belief $b(x_t) \triangleq p(x_t|u_{1:t}, z_{1:t})$ is maintained over all its possible locations $x_t$ where $u_{1:t} \triangleq (u_1, \ldots, u_t)^\top$ and $z_{1:t} \triangleq (z_1, \ldots, z_t)^\top$ denote, respectively, column vectors of past control actions performed and realized field measurements taken by the robot up until time step $t$. To track such a belief, after the robot has performed an action $u_t$ and taken a realized measurement $z_t$ at each time step $t$, the Bayes filter updates the prior belief $b(x_{t-1})$ of the robot's location to the posterior belief $b(x_t) = \beta p(z_t|x_t) \int p(x_t|u_t, x_{t-1})b(x_{t-1})\mathrm{d}x_{t-1}$ where $1/\beta$ is a normalizing constant, $p(x_t|u_t, x_{t-1})$ is a *motion model* representing the probability of the robot moving from locations $x_{t-1}$ to $x_t$ after performing action $u_t$, and $p(z_t|x_t)$ is an *observation model* describing the likelihood of taking realized measurement $z_t$ at location $x_t$.

To preserve efficiency, the Bayes filter imposes a Markov property on the observation model: Given the robot's current location $x_t$, its current measurement $z_t$ is conditionally

---

[2]The field measurement $z_t$ is indexed by time step $t$ instead of the corresponding location $x_t$ since $x_t$ is not known to the robot.

independent of past actions $u_{1:t}$ and measurements $z_{1:t-1}$:

$$p(z_t|x_t, u_{1:t}, z_{1:t-1}) = p(z_t|x_t) . \quad (7)$$

In other words, the robot's past actions performed and measurements taken during its exploration and localization are not exploited for learning the observation model. This is conventionally assumed by existing works either representing the observation model using a parametric model with known parameters (Thrun, Burgard, and Fox 2005) or training it offline using prior training data. The disadvantages of the former are extensively discussed by Ko and Fox (2009a) while that of the latter are already detailed in Section 1.

In the case of multiple fields (say, $M$ of them), let $z_t^m$ denote the realized measurement taken from field $m$ at location $x_t$ for $m = 1, \ldots, M$. Then, the observation model becomes $p(z_t^1, \ldots, z_t^M|x_t) = \prod_{m=1}^{M} p(z_t^m|x_t)$ such that the equality follows from an assumption of independence of measurements between fields to ease computations.

## 3 Online Sparse GP Observation Model

In contrast to existing works discussed in Section 1, our GP-Localize algorithm does not need to impose the restrictive Markov property (7) on the observation model, which can then be derived by marginalizing out the random locations visited and observed by the robot up until time step $t - 1$:

$$p(z_t|x_t, u_{1:t}, z_{1:t-1}) \quad (8)$$

$$= \eta \int b(x_0) \prod_{i=1}^{t} p(x_i|u_i, x_{i-1})p(z_t|x_t, x_{1:t-1}, z_{1:t-1})\mathrm{d}x_{0:t-1}$$

where $1/\eta = p(x_t|u_{1:t}, z_{1:t-1})$ is a normalizing constant, $b(x_0) = p(x_0)$ is the belief of the robot's initial location at time step 0, $x_{1:t-1} \triangleq \{x_1, \ldots, x_{t-1}\}$ denotes a set of locations visited and observed by the robot up until time step $t - 1$, and $p(z_t|x_t, x_{1:t-1}, z_{1:t-1}) = \mathcal{N}(\mu_{x_t|x_{1:t-1}}, \sigma_{x_tx_t|x_{1:t-1}})$ is a Gaussian predictive distribution provided by the GP (Section 2.1). The derivation of (8) is in (Xu et al. 2014).

To make computations tractable but not constrain the type of motion model that can be specified, the observation model (8) is approximated using Monte Carlo integration:

$$p(z_t|x_t, u_{1:t}, z_{1:t-1}) \approx \frac{1}{C} \sum_{c=1}^{C} p(z_t|x_t, x_{1:t-1}^c, z_{1:t-1}) \quad (9)$$

where $x_{1:t-1}^c$ denotes a $c$-th sample path simulated by first drawing the robot's initial location $x_0^c$ from $b(x_0)$ and then sampling $x_i^c$ from motion model $p(x_i|u_i, x_{i-1}^c)$ for $i = 1, \ldots, t - 1$ given its past actions $u_{1:t-1}$ while ensuring $p(x_t|u_t, x_{t-1}^c) > 0$, as observed in (8). For a practical implementation, instead of re-simulating the entire sample paths (hence, incurring linear time in $t$) at each time step, each $c$-th sample path is incrementally updated from $x_{1:t-2}^c$ (i.e., obtained in previous time step) to $x_{1:t-1}^c$ (i.e., needed in current time step) by including $x_{t-1}^c$ sampled from motion model $p(x_{t-1}|u_t, x_{t-2}^c)$ without accounting for motion constraint $p(x_t|u_t, x_{t-1}^c) > 0$. As a result, the time spent in incrementally updating the $C$ sample paths at each time step is independent of $t$. To mitigate the effect of ignoring the constraint, we introduce a strategy in Remark 3 after Theorem 1 that exploits a structural property of our proposed online sparse GP. In practice, such an implementation yields considerable

time savings (i.e., time independent of $t$) and does not result in poor localization performance empirically (Section 4).

The scalability of our GP-Localize algorithm therefore depends on whether the Gaussian predictive probability $p(z_t|x_t, x_{1:t-1}^c, z_{1:t-1})$ in (9) can be derived efficiently. Computing it with full GP, PITC, or FITC (Section 2) directly incurs, respectively, $\mathcal{O}(t^3)$, $\mathcal{O}(t(|\mathcal{S}|^2 + (t/N)^2))$, and $\mathcal{O}(t|\mathcal{S}|^2)$ time. Since $t$ is expected to be large for persistent localization, it is computationally impractical to use these offline full GP and sparse GP approximation methods to repeatedly train the observation model at each filtering step. Even when the online GP proposed by Csató and Opper (2002) is used, it still incurs quadratic time in $t$ per filtering step. In the following subsection, we will propose an online sparse GP that can achieve constant time (i.e., independent of $t$) at each filtering step $t$.

### 3.1 Online Sparse GP Approximation

The key idea underlying our proposed online sparse GP is to summarize the newly gathered data/observations at regular time intervals/slices, assimilate the summary information of the new data with that of all the previously gathered data/observations, and then exploit the resulting assimilated summary information to compute the Gaussian predictive probability $p(z_t|x_t, x_{1:t-1}^c, z_{1:t-1})$ in (9). The details of our proposed online sparse GP will be described next.

Let each time slice $n$ span time/filtering steps $(n-1)\tau + 1$ to $n\tau$ for some user-defined slice size $\tau \in \mathbb{Z}^+$ and the number of time slices available thus far up until time step $t$ be denoted by $N$ (i.e., $N\tau < t$).

**Definition 1 (Slice Summary)** *Given a support set $\mathcal{S} \subset \mathcal{X}$ common to all $C$ sample paths, the subset $\mathcal{D}_n \triangleq x_{(n-1)\tau+1:n\tau}^c$ of the $c$-th sample path $x_{1:t-1}^c$ simulated during the time slice $n$, and the column vector $z_{\mathcal{D}_n} = z_{(n-1)\tau+1:n\tau}$ of corresponding realized measurements taken by the robot, the slice summary of time slice $n$ is defined as a tuple $(\mu_{\circledS}^n, \Sigma_{\circledS}^n)$ for $n = 1, \ldots, N$ where*

$$\mu_{\circledS}^n \triangleq \Sigma_{\mathcal{S}\mathcal{D}_n} \Sigma_{\mathcal{D}_n\mathcal{D}_n|\mathcal{S}}^{-1}(z_{\mathcal{D}_n} - \mu_{\mathcal{D}_n})$$

$$\Sigma_{\circledS}^n \triangleq \Sigma_{\mathcal{S}\mathcal{D}_n} \Sigma_{\mathcal{D}_n\mathcal{D}_n|\mathcal{S}}^{-1} \Sigma_{\mathcal{D}_n\mathcal{S}}$$

*such that $\mu_{\mathcal{D}_n}$ is defined in a similar manner as $\mu_{\mathcal{D}}$ in (1) and $\Sigma_{\mathcal{D}_n\mathcal{D}_n|\mathcal{S}}$ is a posterior covariance matrix with components $\sigma_{xx'|\mathcal{S}}$ for all $x, x' \in \mathcal{D}_n$, each of which is defined in a similar way as (4).*

*Remark.* The support set $\mathcal{S} \subset \mathcal{X}$ of locations does not have to be observed because the slice summary is independent of $z_{\mathcal{S}}$. So, the support set $\mathcal{S}$ can be selected *prior* to exploration and localization from $\mathcal{X}$ using an offline greedy active learning algorithm such as (Krause, Singh, and Guestrin 2008).

**Definition 2 (Assimilated Summary)** *Given $(\mu_{\circledS}^n, \Sigma_{\circledS}^n)$, the assimilated summary $(\mu_{@}^n, \Sigma_{@}^n)$ of time slices $1$ to $n$ is updated from the assimilated summary $(\mu_{@}^{n-1}, \Sigma_{@}^{n-1})$ of time slices $1$ to $n-1$ using $\mu_{@}^n \triangleq \mu_{@}^{n-1} + \mu_{\circledS}^n$ and $\Sigma_{@}^n \triangleq \Sigma_{@}^{n-1} + \Sigma_{\circledS}^n$ for $n = 1, \ldots, N$ where $\mu_{@}^0 \triangleq 0$ and $\Sigma_{@}^0 \triangleq \Sigma_{\mathcal{S}\mathcal{S}}$.*

*Remark* 1. After constructing and assimilating $(\mu_{\circledS}^n, \Sigma_{\circledS}^n)$ with $(\mu_{@}^{n-1}, \Sigma_{@}^{n-1})$ to form $(\mu_{@}^n, \Sigma_{@}^n)$, $\mathcal{D}_n = x_{(n-1)\tau+1:n\tau}^c$,

$z_{\mathcal{D}_n} = z_{(n-1)\tau+1:n\tau}$, and $(\mu_{\circledS}^n, \Sigma_{\circledS}^n)$ (Definition 1) are no longer needed and can be removed from memory. As a result, at time step $t$ where $N\tau + 1 \leq t \leq (N+1)\tau$, only $(\mu_{@}^N, \Sigma_{@}^N)$, $x_{N\tau+1:t-1}^c$, and $z_{N\tau+1:t-1}$ have to be kept in memory, thus requiring only constant memory (i.e., independent of $t$).

*Remark* 2. The slice summaries are constructed and assimilated at a regular time interval of $\tau$, specifically, at time steps $N\tau + 1$ for $N \in \mathbb{Z}^+$.

**Theorem 1** *Given $\mathcal{S} \subset \mathcal{X}$ and $(\mu_{@}^N, \Sigma_{@}^N)$, our online sparse GP computes a Gaussian predictive distribution $p(z_t|x_t, \mu_{@}^N, \Sigma_{@}^N) = \mathcal{N}(\widetilde{\mu}_{x_t}, \widetilde{\sigma}_{x_t x_t})$ of the measurement at any location $x_t \in \mathcal{X}$ at time step $t$ (i.e., $N\tau + 1 \leq t \leq (N+1)\tau$) with the following posterior mean and variance:*

$$\widetilde{\mu}_{x_t} \triangleq \mu_{x_t} + \Sigma_{x_t\mathcal{S}} \left(\Sigma_{@}^N\right)^{-1} \mu_{@}^N \tag{10}$$

$$\widetilde{\sigma}_{x_t x_t} \triangleq \sigma_{x_t x_t} - \Sigma_{x_t\mathcal{S}} \left(\Sigma_{\mathcal{S}\mathcal{S}}^{-1} - \left(\Sigma_{@}^N\right)^{-1}\right) \Sigma_{\mathcal{S}x_t}. \tag{11}$$

*If $t = N\tau + 1$, $\widetilde{\mu}_{x_t} = \mu_{x_t|x_{1:t-1}^c}^{\text{PITC}}$ and $\widetilde{\sigma}_{x_t x_t} = \sigma_{x_t x_t|x_{1:t-1}^c}^{\text{PITC}}$.*

Its proof is given in (Xu et al. 2014).

*Remark* 1. Theorem 1 implies that our proposed online sparse GP is in fact equivalent to an online learning formulation/variant of the offline PITC (Section 2.2). Supposing $\tau < |\mathcal{S}|$, the $\mathcal{O}(t|\mathcal{S}|^2)$ time incurred by offline PITC to compute $p(z_t|x_t, x_{1:t-1}^c, z_{1:t-1})$ in (9) can then be reduced to $\mathcal{O}(\tau|\mathcal{S}|^2)$ time (i.e., time independent of $t$) incurred by our online sparse GP at time steps $t = N\tau + 1$ for $N \in \mathbb{Z}^+$ when slice summaries are constructed and assimilated. Otherwise, our online sparse GP only incurs $\mathcal{O}(|\mathcal{S}|^2)$ time per time step.

*Remark* 2. The above equivalence result allows the structural property of our online sparse GP to be elucidated using that of offline PITC: The measurements $Z_{\mathcal{D}_1}, \ldots, Z_{\mathcal{D}_N}, Z_{x_t}$ between different time slices are assumed to be conditionally independent given $Z_{\mathcal{S}}$. Such an assumption enables the data gathered during each time slice to be summarized independently of that in other time slices. Increasing slice size $\tau$ (i.e., less frequent assimilations of larger slice summaries) relaxes this conditional independence assumption (hence, potentially improving the fidelity of the resulting observation model), but incurs more time at time steps when slice summaries are constructed and assimilated (see Remark 1).

*Remark* 3. Recall (see paragraph after (9)) that the motion constraint $p(x_t|u_t, x_{t-1}^c) > 0$ is not accounted for when sampling $x_{t-1}^c$ from motion model $p(x_{t-1}|u_t, x_{t-2}^c)$ at each time step $t$. To mitigate the effect of ignoring the constraint, at time steps $t = N\tau + 2$ for $N \in \mathbb{Z}^+$, we draw $x'_{N\tau}$ from the particle-based belief $b(x_{N\tau})$ maintained in our experiments (Section 4) and use it (instead of $x_{N\tau}^c$) for sampling $x_{N\tau+1}^c$ from motion model $p(x_{N\tau+1}|u_i, x'_{N\tau})$. Doing this at a regular time interval of $\tau$ reduces the deviation of the simulated sample paths from the particle-based beliefs updated at each time step and consequently allows the sample paths to satisfy the motion constraint more often, especially when $\tau$ is small. Such a strategy may cause the sampled $x_{N\tau+1}^c$ not to be located close to $x_{N\tau}^c$ (hence, their corresponding realized measurements are less spatially correlated) since $x_{N\tau+1}^c$ is not sampled from motion model $p(x_{N\tau+1}|u_i, x_{N\tau}^c)$. But,

this occurs at a lower frequency of $1/\tau$, as compared to not considering the motion constraint at every time step. Furthermore, this fits well with the structural property of our online sparse GP that assumes $Z_{N\tau}$ and $Z_{N\tau+1}$ (or, more generally, $Z_{\mathcal{D}_N}$ and $Z_{x_t}$) to be conditionally independent.

*Remark* 4. Since offline PITC generalizes offline FITC (Section 2.2), our online sparse GP generalizes the online learning variant of FITC (i.e., $\tau = 1$) (Csató and Opper 2002)[3].

When $N\tau + 1 < t \leq (N+1)\tau$ (i.e., before the next slice summary of time slice $N + 1$ is constructed and assimilated), the most recent observations (i.e., $\mathcal{D}' \triangleq x^c_{N\tau+1:t-1}$ and $z_{\mathcal{D}'} = z_{N\tau+1:t-1}$), which are often highly informative, are not used to update $\widetilde{\mu}_{x_t}$ (10) and $\widetilde{\sigma}_{x_t x_t}$ (11). This will hurt the localization performance, especially when $\tau$ is large and the robot is localizing in an unexplored area with little/no observations; the field within this area thus cannot be predicted well with the current assimilated summary. To resolve this, we exploit incremental update formulas of Gaussian posterior mean and variance (Xu et al. 2014) to update $\widetilde{\mu}_{x_t}$ and $\widetilde{\sigma}_{x_t x_t}$ with the most recent observations, thereby yielding a Gaussian predictive distribution $p(z_t|x_t, \mu^N_{@}, \Sigma^N_{@}, \mathcal{D}', z_{\mathcal{D}'}) = \mathcal{N}(\widetilde{\mu}_{x_t|\mathcal{D}'}, \widetilde{\sigma}_{x_t x_t|\mathcal{D}'})$ where

$$\widetilde{\mu}_{x_t|\mathcal{D}'} \triangleq \widetilde{\mu}_{x_t} + \widetilde{\Sigma}_{x_t \mathcal{D}'} \widetilde{\Sigma}^{-1}_{\mathcal{D}' \mathcal{D}'} (z_{\mathcal{D}'} - \widetilde{\mu}_{\mathcal{D}'}) \quad (12)$$

$$\widetilde{\sigma}_{x_t x_t|\mathcal{D}'} \triangleq \widetilde{\sigma}_{x_t x_t} - \widetilde{\Sigma}_{x_t \mathcal{D}'} \widetilde{\Sigma}^{-1}_{\mathcal{D}' \mathcal{D}'} \widetilde{\Sigma}_{\mathcal{D}' x_t} \quad (13)$$

such that $\widetilde{\mu}_{\mathcal{D}'}$ is a column vector with mean components $\widetilde{\mu}_x$ (i.e., defined similarly to (10)) for all $x \in \mathcal{D}'$, $\widetilde{\Sigma}_{x_t \mathcal{D}'}$ is a row vector with covariance components $\widetilde{\sigma}_{x_t x}$ (i.e., defined similarly to (11)) for all $x \in \mathcal{D}'$, $\widetilde{\Sigma}_{\mathcal{D}' x_t}$ is the transpose of $\widetilde{\Sigma}_{x_t \mathcal{D}'}$, and $\widetilde{\Sigma}_{\mathcal{D}' \mathcal{D}'}$ is a matrix with covariance components $\widetilde{\sigma}_{xx'}$ (i.e., defined similarly to (11)) for all $x, x' \in \mathcal{D}'$.

**Theorem 2** *Computing* $p(z_t|x_t, \mu^N_{@}, \Sigma^N_{@}, \mathcal{D}', z_{\mathcal{D}'})$ *(i.e., (12) and (13)) incurs* $\mathcal{O}(\tau|\mathcal{S}|^2)$ *time at time steps* $t = N\tau + 1$ *for* $N \in \mathbb{Z}^+$ *and* $\mathcal{O}(|\mathcal{S}|^2)$ *time otherwise. It requires* $\mathcal{O}(|\mathcal{S}|^2)$ *memory at each time step.*

Its proof is given in (Xu et al. 2014). Theorem 2 indicates that our online sparse GP incurs constant time and memory (i.e., independent of $t$) per time step.

## 4 Experiments and Discussion

This section evaluates the localization performance, time efficiency, and scalability of our GP-Localize algorithm empirically through simulated experiments with three real-world datasets: (a) *Wireless signal strength* (WSS) (signal-to-noise ratio) data (Chen and Guestrin 2007) produced by 6 WiFi access points (APs) and measured at over 200 locations throughout the fifth floor of Wean Hall in Carnegie Mellon University (Fig. 1, Section 4.1), (b) *indoor environmental quality* (IEQ) (i.e., temperature (°F) and light (Lux)) data (Bodik et al. 2004) measured by 54 sensors deployed in the Intel Berkeley Research lab (Fig. 3, Section 4.2), (c) *urban traffic speeds* (UTS) (km/h) data (Chen et al. 2012; 2013) measured at 775 road segments (including highways, arterials, slip roads, etc.) of an urban road network

in Tampines area, Singapore during evening peak hours on April 20, 2011 with a mean speed of 47.6 km/h and a standard deviation of 20.5 km/h (Fig. 4a, Section 4.3), and (d) a real Pioneer 3-DX mobile robot (i.e., mounted with a weather board) experiment on a trajectory of about 280 m in the *Singapore-MIT Alliance for Research and Technology Future Urban Mobility* (SMART FM) IRG office/lab gathering 561 relative light (%) observations/data for GP localization (Fig. 5, Section 4.4). Different from the 2-dimensional spatial domains of the WSS, IEQ, and light fields, each road segment of the urban road network is specified by a 5-dimensional vector of features: length, number of lanes, speed limit, direction, and time. The UTS field is modeled using a relational GP (previously developed in (Chen et al. 2012)) whose correlation structure can exploit both the road segment features and road network topology information. The hyperparameters of each GP modeling a different field are learned using the data via maximum likelihood estimation (Rasmussen and Williams 2006). Our GP-Localize algorithm is implemented using an odometry motion model[4], our online sparse GP (i.e., setting $\tau = 10$ and $|\mathcal{S}| = 40$) for representing the observation model (Section 3), and a particle filter[4] of 400 particles for representing the belief of the robot's location. The number $C$ of sample paths in (9) is set to 400 for all experiments. For the simulated experiments with the WSS and IEQ data, the control actions (i.e., odometry information) are generated using the realistic Pioneer mobile robot module in Player/Stage simulator (Gerkey, Vaughan, and Howard 2003) and the measurements taken along the generated trajectory of 421 (336) time steps from the WSS (IEQ) fields shown in Fig. 1 (Fig. 2) are the Gaussian predictive/posterior means (1) of each full GP modeling a separate field trained using the data. For the simulated experiment with the UTS data, the control actions of the mobile probe vehicle are assumed not to be known; its transition probability of moving from one road segment to another can be learned from vehicle route data using the hierarchical Bayesian nonparametric approach of Yu et al. (2012). The measurements taken along its generated trajectory of 370 time steps from the UTS field are shown in Fig. 4.

The localization performance/error (i.e., distance between the robot's estimated and true locations) and scalability of our GP-Localize algorithm is compared to that of two sparse GP localization algorithms: (a) The *SoD-Truncate* method uses $|\mathcal{S}| = 10$ most recent observations (i.e., compared to $|\mathcal{D}'| < \tau = 10$ most recent observations considered by our online sparse GP besides the assimilated summary) as training data at each filtering step while (b) the *SoD-Even* method uses $|\mathcal{S}| = 40$ observations (i.e., compared to the support set of $|\mathcal{S}| = 40$ possibly unobserved locations selected *prior* to localization and exploited by our online sparse GP) evenly distributed over the time of localization. The scalability of GP-Localize is further compared to that of GP localization algorithms employing full GP and offline PITC (Section 2).

---

[3]Snelson (2007) pointed out that the sparse online GP of Csató and Opper (2002) is an online learning variant of offline FITC.

[4]Due to lack of space, an interested reader is referred to (Thrun, Burgard, and Fox 2005) for the technical details of the odometry motion model and particle filter.
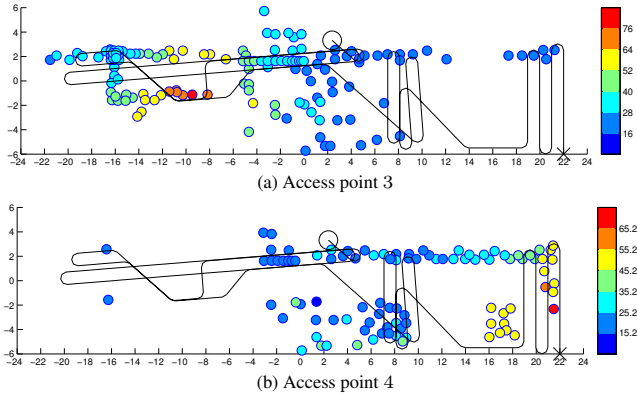
(a) Access point 3



(b) Access point 4

Figure 1: WSS (signal-to-noise ratio) data produced by WiFi APs 3 and 4 and measured at locations denoted by small colored circles. Robot trajectory starts at '×' and ends at '◯'. WSS data produced by WiFi APs 1, 2, 5, 6 are shown in (Xu et al. 2014).

## 4.1 Wireless Signal Strength (WSS) Fields

Table 1 shows the localization errors (no units given in WSS data) of GP-localize, SoD-Truncate, and SoD-Even averaged over all 421 time steps and 5 simulation runs. It can be observed that GP-Localize outperforms the other two methods in every single field and in multiple fields (i.e., all 6): The observation model (i.e., represented by our online sparse GP) of GP-Localize can exploit the assimilated summary and the most recent observations to predict, respectively, the fields in explored and unexplored areas better. In contrast, SoD-Truncate performs poorly in explored areas since its predictive capability is limited by using only the most recent observations. The limited observations of SoD-Even can only cover the entire area sparsely, thus producing an inaccurate observation model.

It can also be observed from Table 1 that GP-Localize achieves its largest (smallest) localization error in field 3 (4): Fig. 1a shows that the robot does not explore the area on the left with highly varying measurements well enough, thus yielding an assimilated summary that is less informative to localization in this area. Though it explores the area on the right densely, the field in this area is relatively constant, hence making localization difficult. As a result, localization error is high in field 3. On the other hand, Fig. 1b shows that the robot explores the area on the right with highly varying measurements densely, thus achieving low error in field 4.

Fig. 2a shows the localization error of GP-Localize at each time step in every single field and in multiple fields (i.e., all 6) averaged over 5 runs. It can be observed that although the error in multiple fields is not always smallest at each time step, it often tends to be close to (if not, lower than) the lowest error among all single fields and, more importantly, is not so high like those in single fields 1, 2, 3, 5, 6

| Field | 1 | 2 | 3 | 4 | 5 | 6 | Multiple |
|---|---|---|---|---|---|---|---|
| **GP-Localize** | 17.0 | 17.1 | 19.4 | 8.2 | 17.6 | 17.0 | 6.7 |
| SoD-Truncate | 25.0 | 23.1 | 23.4 | 21.4 | 21.0 | 22.4 | 20.1 |
| SoD-Even | 21.2 | 20.5 | 22.9 | 21.9 | 21.5 | 21.5 | 21.8 |

Table 1: Localization errors in single WSS fields 1, 2, 3, 4, 5, 6 corresponding to APs 1, 2, 3, 4, 5, 6 and in multiple fields.
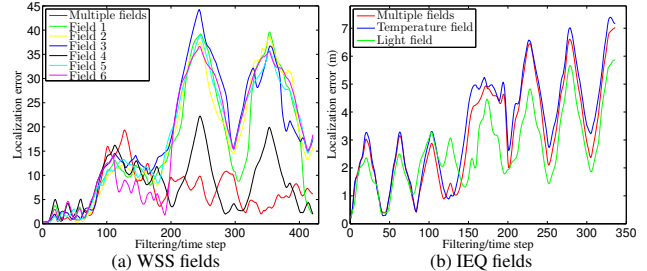


(a) WSS fields     (b) IEQ fields

Figure 2: Graphs of localization error vs. no. of time steps.
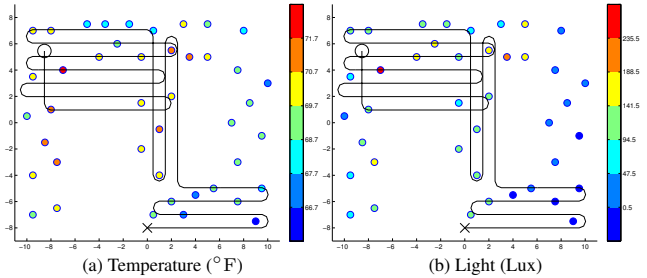


(a) Temperature (°F)     (b) Light (Lux)

Figure 3: Temperature (°F) and light (Lux) data measured at sensor locations denoted by small colored circles. Robot trajectory starts at '×' and ends at '◯'. Each unit along the vertical and horizontal axes is 1 m.

after 200 time steps. In practice, since it is usually not known which single field yields a low or high error at each time step, a more robust GP-Localize algorithm (i.e., achieved by exploiting multiple fields) is preferred.

## 4.2 Indoor Environmental Quality (IEQ) Fields

Table 2 shows the localization errors (m) of the tested methods averaged over all 336 time steps and 5 simulation runs. Similar to the observations for WSS fields (Section 4.1), GP-Localize outperforms the other two methods in every single field and in multiple fields, as explained in the previous section. It can also be observed that GP-Localize achieves a smaller error in the light field than in the temperature field because the measurements of the light field vary slightly more than that of the temperature field, as shown in Fig. 3. Fig. 2b shows the error of GP-Localize at each time step in every single field and in multiple fields averaged over 5 runs. It can again be observed that although the error in multiple fields is not always smallest at each time step, it is often close to (if not, lower than) the lowest error among all single fields and not as high as that in the single temperature field. Our GP-Localize algorithm exploiting multiple fields is therefore more robust in this experiment.

## 4.3 Urban Traffic Speeds (UTS) Field

For the UTS field, the localization error is defined as the geodesic (i.e., shortest path) distance between the vehicle's

| Field | Temperature | Light | Multiple |
|---|---|---|---|
| **GP-Localize** | 3.5 | 2.7 | 3.2 |
| SoD-Truncate | 4.0 | 6.6 | 3.8 |
| SoD-Even | 5.3 | 3.3 | 4.4 |

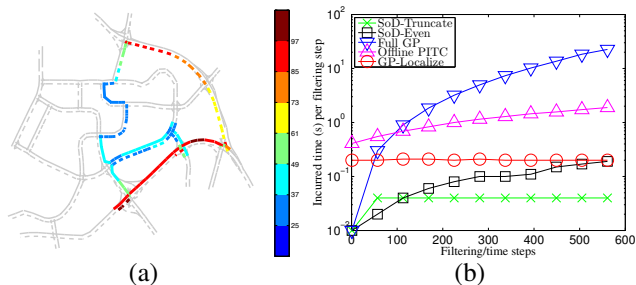Table 2: Localization errors (m) in single and multiple IEQ fields.

Figure 4: (a) UTS (km/h) data taken along the vehicle trajectory denoted by colored road segments of the urban road network in Tampines area, Singapore, and (b) graphs of incurred time (s) per filtering step vs. no. of filtering/time steps comparing different GP localization algorithms in the real Pioneer 3-DX mobile robot experiment.
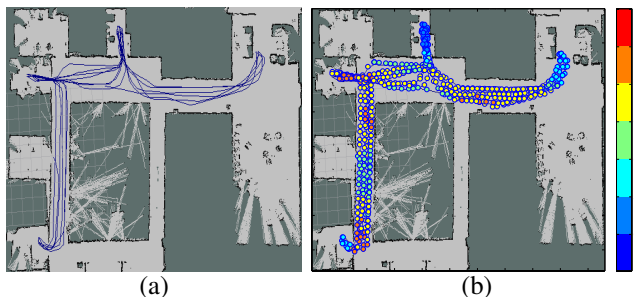


Figure 5: (a) Pioneer 3-DX mobile robot trajectory of about 280 m in the SMART FM IRG office/lab generated by AMCL package in ROS, along which (b) 561 relative light (%) observations/data are gathered at locations denoted by small colored circles.

estimated and true residing road segments with respect to the road network topology (Fig. 4). GP-Localize, SoD-Truncate, and SoD-Even achieve, respectively, localization errors of 2.8, 7.3, and 6.2 road segments averaged over all 370 time steps and 3 simulation runs.

### 4.4 Real Pioneer 3-DX Mobile Robot Experiment

The *adaptive Monte Carlo localization* (AMCL) package in the *Robot Operating System* (ROS) is run on a Pioneer 3-DX mobile robot mounted with a SICK LMS200 laser rangefinder to determine its trajectory (Fig. 5a) and the 561 locations at which the relative light measurements are taken (Fig. 5b); these locations are assumed to be the ground truth. GP-Localize, SoD-Truncate, and SoD-Even achieve, respectively, localization errors of 2.1 m, 5.4 m, and 4.6 m averaged over all 561 time steps and 3 runs.

Fig. 4b shows the time incurred by GP-Localize, SoD-Truncate, SoD-Even, full GP, and offline PITC at each time step using 100 particles averaged over 5 runs. It can be seen that, with more time steps, the time incurred by full GP, offline PITC, and SoD-Even increase while that of GP-Localize and SoD-Truncate remain constant. GP-Localize is clearly much more scalable (i.e., constant time) in $t$ than full GP and offline PITC. Though it incurs slightly more time than SoD-Truncate and SoD-Even, it can localize significantly better (Sections 4.1 and 4.2).

## 5   Conclusion

This paper describes the GP-Localize algorithm for persistent robot localization whose observation model is represented by our novel online sparse GP, thus achieving constant time and memory (i.e., independent of the size of the data) per filtering step. We theoretically analyze the equivalence of our online sparse GP to the online learning variant of offline PITC. We empirically demonstrate that GP-Localize outperforms existing GP localization algorithms in terms of localization performance and scalability and achieves robustness by exploiting multiple fields. Besides using our online sparse GP for persistent robot localization, note that it can in fact be applied to a wide variety of applications and are especially desirable (i.e., due to runtime and memory being independent of the size of data) for tasks with data streaming in over time or real-time requirements. Some robotic tasks include adaptive sampling, information gathering, learning of robot arm control (Low, Leow, and Ang, Jr. 2002a; 2002b; 2005), visual tracking and head pose estimation for human-robot interaction. For non-robotic applications, they include traffic and weather prediction, online action recognition, online recommendation systems, online classification, among others.

A limitation of GP-Localize, as observed in our experiments, is that it does not localize well in near-constant fields, which is expected. So, in our future work, we plan to generalize our algorithm to handle richer, high-dimensional sensing data like laser scans and camera images (Natarajan et al. 2012a; 2012b; Natarajan, Low, and Kankanhalli 2014). We also like to investigate the effect of varying the slice size $\tau$ on the localization error of GP-Localize empirically and remove the assumption of independence between fields by exploiting techniques like multi-output GPs and co-kriging for modeling their correlation. Lastly, as mentioned in Section 4, the hyperparameters of each GP are learned using the data by maximizing the log marginal likelihood. The sparse approximation method employed by offline PITC to improve the scalability of the full GP can be similarly applied to computing such a log marginal likelihood scalably, as explained in (Quiñonero-Candela and Rasmussen 2005) (i.e., equation 30 in Section 9). Since our online sparse GP is the online variant of the offline PITC, the log marginal likelihood can be computed and maximized in an online manner as well. The exact details will be specified in the future extension of this work.

## References

Bodik, P.; Guestrin, C.; Hong, W.; Madden, S.; Paskin, M.; and Thibaux, R. 2004. http://www.select.cs.cmu.edu/data/labapp3/index.html.

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

Cao, N.; Low, K. H.; and Dolan, J. M. 2013. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proc. AAMAS*, 7–14.

Chen, K., and Guestrin, C. 2007. http://www.select.cs.cmu.edu/data/index.html.

Chen, J.; Low, K. H.; Tan, C. K.-Y.; Oran, A.; Jaillet, P.; Dolan, J. M.; and Sukhatme, G. S. 2012. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, 163–173.

Chen, J.; Cao, N.; Low, K. H.; Ouyang, R.; Tan, C. K.-Y.; and Jaillet, P. 2013. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, 152–161.

Chen, J.; Low, K. H.; and Tan, C. K.-Y. 2013. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*.

Csató, L., and Opper, M. 2002. Sparse online Gaussian processes. *Neural Computation* 14(2):641–669.

Dolan, J. M.; Podnar, G.; Stancliff, S.; Low, K. H.; Elfes, A.; Higinbotham, J.; Hosler, J. C.; Moisan, T. A.; and Moisan, J. 2009. Cooperative aquatic sensing using the telesupervised adaptive ocean sensor fleet. In *Proc. SPIE Conference on Remote Sensing of the Ocean, Sea Ice, and Large Water Regions*, volume 7473.

Ferris, B.; Fox, D.; and Lawrence, N. 2007. WiFi-SLAM using Gaussian process latent variable models. In *Proc. IJCAI*, 2480–2485.

Ferris, B.; Hähnel, D.; and Fox, D. 2006. Gaussian processes for signal strength-based location estimation. In *Proc. RSS*.

Gerkey, B. P.; Vaughan, R. T.; and Howard, A. 2003. The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proc. ICAR*, 317–323.

Hoang, T. N.; Low, K. H.; Jaillet, P.; and Kankanhalli, M. 2014. Nonmyopic $\epsilon$-Bayes-optimal active learning of Gaussian processes. In *Proc. ICML*.

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

Ko, J., and Fox, D. 2009b. Learning GP-BayesFilters via Gaussian process latent variable models. In *Proc. RSS*.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR* 9:235–284.

Low, K. H.; Gordon, G. J.; Dolan, J. M.; and Khosla, P. 2007. Adaptive sampling for multi-robot wide-area exploration. In *Proc. IEEE ICRA*, 755–760.

Low, K. H.; Chen, J.; Dolan, J. M.; Chien, S.; and Thompson, D. R. 2012. Decentralized active robotic exploration and mapping for probabilistic field classification in environmental sensing. In *Proc. AAMAS*, 105–112.

Low, K. H.; Dolan, J. M.; and Khosla, P. 2008. Adaptive multi-robot wide-area exploration and mapping. In *Proc. AAMAS*, 23–30.

Low, K. H.; Dolan, J. M.; and Khosla, P. 2009. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In *Proc. ICAPS*, 233–240.

Low, K. H.; Dolan, J. M.; and Khosla, P. 2011. Active Markov information-theoretic path planning for robotic environmental sensing. In *Proc. AAMAS*, 753–760.

Low, K. H.; Leow, W. K.; and Ang, Jr., M. H. 2002a. A hybrid mobile robot architecture with integrated planning and control. In *Proc. AAMAS*, 219–226.

Low, K. H.; Leow, W. K.; and Ang, Jr., M. H. 2002b. Integrated planning and control of mobile robot with self-organizing neural network. In *Proc. IEEE ICRA*, 3870–3875.

Low, K. H.; Leow, W. K.; and Ang, Jr., M. H. 2005. An ensemble of cooperative extended Kohonen maps for complex robot motion tasks. *Neural Comput.* 17(6):1411–1445.

Natarajan, P.; Hoang, T. N.; Low, K. H.; and Kankanhalli, M. 2012a. Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance. In *Proc. AAMAS*, 155–162.

Natarajan, P.; Hoang, T. N.; Low, K. H.; and Kankanhalli, M. 2012b. Decision-theoretic coordination and control for active multi-camera surveillance in uncertain, partially observable environments. In *Proc. ICDSC*.

Natarajan, P.; Low, K. H.; and Kankanhalli, M. 2014. Decision-theoretic approach to maximizing fairness in multi-target observation in multi-camera surveillance. In *Proc. AAMAS*.

Ouyang, R.; Low, K. H.; Chen, J.; and Jaillet, P. 2014. Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena. In *Proc. AAMAS*.

Podnar, G.; Dolan, J. M.; Low, K. H.; and Elfes, A. 2010. Telesupervised remote surface water quality sensing. In *Proc. IEEE Aerospace Conference*.

Quiñonero-Candela, J., and Rasmussen, C. E. 2005. A unifying view of sparse approximate Gaussian process regression. *JMLR* 6:1939–1959.

Rasmussen, C. E., and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Snelson, E. L. 2007. *Flexible and efficient Gaussian process models for machine learning*. Ph.D. Thesis, University College London, London, UK.

Thrun, S.; Burgard, W.; and Fox, D. 2005. *Probabilistic Robotics*. Cambridge, MA: MIT Press.

Xu, N.; Low, K. H.; Chen, J.; Lim, K. K.; and Özgül, E. B. 2014. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. arXiv:1404.5165.

Yu, J.; Low, K. H.; Oran, A.; and Jaillet, P. 2012. Hierarchical Bayesian nonparametric approach to modeling and learning the wisdom of crowds of urban traffic route planning agents. In *Proc. IAT*, 478–485.