Short Paper

Point-Wise Fusion of Distributed Gaussian Process Experts (FuDGE) Using a Fully Decentralized Robot Team Operating in Communication-Devoid Environment

Kshitij Tiwari, Sungmoon Jeong, and Nak Young Chong

***Abstract*—In this paper, we focus on large-scale environment monitoring by utilizing a fully decentralized team of mobile robots. The robots utilize the resource constrained-decentralized active sensing scheme to select the most informative (uncertain) locations to observe while conserving allocated resources (battery, travel distance, *etc.*). We utilize a distributed Gaussian process (GP) framework to split the computational load over our fleet of robots. Since each robot is individually generating a model of the environment, there may be conflicting predictions for test locations. Thus, in this paper, we propose an algorithm for aggregating individual prediction models into a single globally consistent model that can be used to infer the overall spatial dynamics of the environment. To make a prediction at a previously unobserved location, we propose a novel gating network for a mixture-of-experts model wherein the weight of an expert is determined by the responsibility of the expert over the unvisited location. The benefit of posing our problem as a centralized fusion with a distributed GP computation approach is that the robots never communicate with each other, individually optimize their own GP models based on their respective observations, and off-load all their learnt models on the base station only at the end of their respective mission times. We demonstrate the effectiveness of our approach using publicly available datasets.**

***Index Terms*—Distributed robot systems, field robots, model fusion, path planning for multiple mobile robot systems, surveillance systems.**

I. Introduction

Modeling the complex dynamics of a large-scale environmental phenomenon is a challenging yet intriguing problem. From a robotics perspective, the challenges arise owing to the fact that the area to be scanned is prohibitively large but the resources allocated to the robots are limited and must be used conservatively. From a machine learning point of view, the choice of representative models itself is a challenge. Many researchers use Gaussian processes (GPs) [1], which are data-driven models, to represent the dynamics of the environment. These models are nonparametric while providing predictive uncertainty bounds. However, there are several limitations associated with GPs, such as poor scalability (cubic in data set size) and a large (quadratic in data set size) memory footprint, which have been widely studied [2]–[4].

Similar works in this domain use the term "fusion" to combine multiple sets of heterogeneous sensor data using GPs as discussed in [5]–[8]. Alternatively, we can use the term "fusion" to define an ensemble of probabilistically fused prediction estimators [2], [3], [9]–[12], which is our area of interest. We refer to two classes of model fusion techniques in this work. First, we refer to product-of-expert (PoE) models like the Bayesian committee machine (BCM) [13] and generalized PoE (gPoE) [9]. In the BCM framework, multiple independent GP experts are trained on subsets of the whole training set, and their confidence is evaluated based on the reduction in uncertainty over the test points. Although this approach is promising in terms of distributing the computational load of a single GP over multiple GP experts, it is not feasible for a real robot implementation. Since this approach works only under the assumption that all GP experts are "jointly trained" such that they "share the same set of hyper-parameters" [2], we cannot apply it directly to a real robotic setup. Doing so for a real robot team would require precise time synchronization and an all-to-all synchronized communication (as was used in the recent work [14]). In the gPoE models from [9], the fusion is carried out over independent GP experts wherein their contributions are determined, *e.g.*, by their differential entropy scores. Both of the above mentioned models are log opinion pool models but the BCM model ensures consistency in the sense that predictions are guaranteed to fall back to the prior when we leave the training data. Second, we refer to mixture-of-expert models [10]–[12] wherein each GP expert specializes in different partitions of the state space and the mixture ensemble automatically allocates the expert, its corresponding specialist zone. This model is a linear opinion pool of experts where the weights are given by input-dependent gating functions. For this, we draw inspiration from the neural network literature [15]–[17], which introduced a point-wise locally weighted fusion technique to evaluate the performance of a predictor over a neighborhood around the probe point. However, these approaches require a sufficiently dense training dataset with access to ground truth. Hence, they cannot be applied directly in a real robotic setup like ours. Beyond the above mentioned solutions, there are other solutions in the literature that deal with multi-agent decentralized exploration like [18] wherein a Dirichlet process mixture of GP experts is used to model a decentralized ensemble of GP experts. In this approach, the requirement of a control parameter α that manages the addition of a new cluster enforces the need of supervision (by base node or human) that can control and instruct a new member to be added to the team when a new cluster is created. Alternative solution proposed for multiple GP experts for decentralized data fusion comes from [3]. This work does not belong to any of the two categories summarized above, and, in this work, the robots share the measurements gathered with their nearest neighbors using consensus filtering. Since in our problem setting we do not allow the robots to communicate with each other neither do we require a supervisor to dynamically add members to the team, thus, we propose a weighted fusion technique suitable for GPs that allows us to evaluate the proximity of a probe point to the training samples of GP experts while evaluating the confidence of each expert.

Manuscript received October 5, 2017; revised January 5, 2018; accepted January 10, 2018. Date of publication February 28, 2018; date of current version June 6, 2018. This paper was recommended for publication by Associate Editor M. Kaess and Editor C. Torras upon evaluation of the reviewers' comments. This work was supported by the Industrial Convergence Core Technology Development Program under Grant 10063172 funded by the Ministry of Trade, Industry and Energy, South Korea. *(Corresponding author: Kshitij Tiwari.)*

The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan (e-mail: kshitij.tiwari@jaist.ac.jp; jeongsm@jaist.ac.jp; nakyoung@jaist.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2018.2794535

In our previous work [19], the distributed GP [2] framework was proposed for a multirobot setup in which multiple robots were allowed to behave as self-dependent GP experts and model the target phenomenon by collecting the most informative training samples from their dedicated training zones. Having done this, we obtained multiple trained GP experts, which made the predicted maps for the spatial dynamics of the target phenomenon. However, we saw that arbitrarily partitioning the space sometimes means that some robots may not have access to sufficient observations to enhance their models. Since each GP expert uses only a subset of the whole training samples, we cannot completely rely on the predictions made by individual GP experts. To overcome these limitations, in our current work, we allow access to the whole sensing area for each robot. The global model is available only at the end of the mission times of all experts. During their respective mission times, there is no communication between the robots, and each robot gathers training samples that are deemed best for improving its own GP model.

We propose a novel algorithm that, at the end of mission times of all robots, fuses all prediction models into a globally consistent model for the entire phenomenon. For this, we fuse the predictions of all GP experts while accounting for the strengths of all the experts to make a fused model, which can perform better than the individual experts. The advantage of our fusion architecture is that it can not only handle heterogeneity in the team in terms of robots themselves (*e.g.,* unmanned aerial vehicles coupled with unmanned ground vehicles), but also any sort of informative path planning can be used for the respective members. Normally, when considering a heterogeneous robot team, different members have different budgets and hence might end their exploration earlier than others. Our architecture can easily account for varying exploratory missions but this is beyond the scope of the current manuscript. Also, our architecture is immune to complete system failures. Since we fuse the maps in a centralized fashion only at the end of all the missions for all robots, should the centralized node fail, our robots would still retain their individual models, which can be fused elsewhere. This, as opposed to the fully centralized architecture of [20], is our first step toward a robust dual-layer architecture while splitting the exploration load over the team and only passing on the map fusion load to the centralized node. We present empirical evaluations backed by statistically significant results to show that our model performs better than existing state-of-the-art models like gPoE and a single GP. A preliminary version of this approach was previously presented as a position paper [21].

II. Problem Formulation

In this section, we introduce the sensing scenario in which we demonstrate our fusion approach and also introduce the notations deemed viable for later use.

A. Sensing Scenario

We begin by explaining the sensing setup. The sensing environment being considered here is similar to the works of [22] albeit there are a few differences and extensions in this work. Our approach serves to extend and enhance the previous works as follows.

1) *Choosing the next-best-location:* Instead of looking at just the immediate neighbors, our approach evaluates all combinations of correlated locations within the field.
2) *Concurrent inference and estimation:* In our approach, we can concurrently *infer* the optimal model parameters by updating the model as and when new data come in, and if required, *estimate* the measurement at any arbitrary input location.
3) *Fusion:* Although the correlated orienteering problem approach [22] is extendable for multirobot setting but it is not clear what happens when multiple robots generate models based on their individual observations. If the setting is a communication-devoid environment like ours, then the robots might end up generating slightly conflicting models of the environment despite gathering overlapping observations. This problem is elegantly resolved by our model where we fuse and generate a globally consistent model of the environment.
4) *Measurement noise:* All measurements gathered are considered to be noisy and the noise variance is itself treated as a parameter to be learnt via inference as opposed to Yu *et al.*'s work [22] where noise-free measurements were considered.
5) *Informativeness of a location:* In [22], the informativeness of a candidate location j was only considered with respect to a specific location i in its immediate neighborhood, independent of the rest. However, in our case, we evaluate the informativeness of a candidate location in terms of the reduction of uncertainty achieved over the entire environment (*i.e.,* all the unobserved locations)
6) *Team size:* Our approach already tries to tackle scenarios where the sensing environment is significantly large and the size of robot team is significantly small to monitor and model the environment given the limited resources.

These extensions are proposed under the following assumptions.

1) The environment to be modeled is quite large and cannot be covered by a single robot.
2) We have a team of robots that cannot communicate with each other to be able to freely cover the vast expanse of the environment of interest.
3) The resources (battery life, flight time, travel distance, *etc.*) available to each member of the team are not enough to perform exhaustive coverage of the phenomenon. Blanket coverage may lead to supreme model performance but is practically infeasible and the robots are tasked with planning budget-limited informative tours to myopically maximize their reward (measured in terms of information gain).
4) The measurements and correlations of the field are not updated during the robot's exploration phase. However, during a new exploration cycle, the measurements are varying in both spatial and temporal domains. Thus, for the scope of this work, we only analyze the spatial domain for a chosen time step.
5) Since the robots are not communicating with each other, they need to maintain their own set of hyper-parameters despite the fact that uncoordinated explorations may lead to more than one robot visiting the same location during exploration. To this end, we will use the term "independent" to refer to individual models maintained by each robot. However, we caution the reader that more than one robot might use the same measurement as a training sample for its respective model, and hence, the term "independent" must not be confused with the term conditional independence in any sense.
6) Most of the environment monitoring datasets only record measurements for static sensors placed at discrete locations. However, not all stations need to be observed at all times. Thus, in this work, we aim to select the "key" locations to be observed. Thus, we no longer have access to static sensors but since the measurements are only available at the locations where the static sensors were previously placed, we restrict our robots to only observe and visit these locations. Thus, in our setting, the locations that can be observed at predefined[1] (similar to [22]).

[1]It must be noted here that GPs can be used to predict measurements at any arbitrary location but since the ground truth for such locations was not made available in the raw dataset, we cannot evaluate the prediction performance and hence were not considered in the current problem setup.

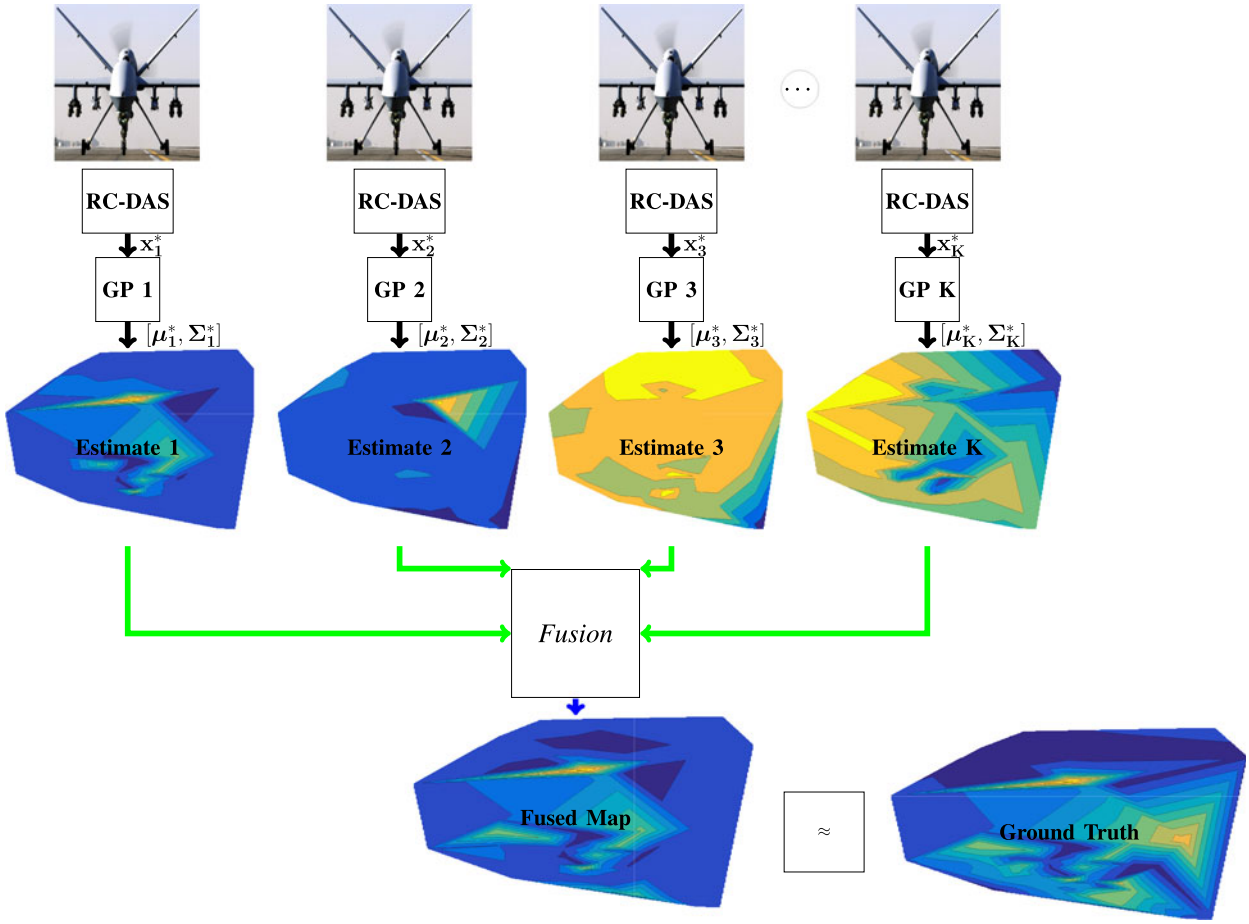

Fig. 1. Sensing scenario. Illustration of the sensing scenario in which the team of mobile robots operates under resource constraints. The aim is to gather optimal observations to make a prediction for the environment defined by posterior mean $\boldsymbol{\mu}_k^*$ and posterior covariance Σ_k^*. Estimates 1–*K* represent the k individualistic prediction maps made by the K robots based on their training samples. x_k^* represents the *next-best-location* chosen by the RC-DAS active sensing for the k^{th} expert. Fused map is the globally consistent fused prediction map generated by using our proposed fusion framework. Our target is to make the fused map as similar to the ground truth as possible. These maps show 2-D spatial interpolation for ease of visualization. In reality, we just have a discrete collection of predicted measurements at predetermined locations.

Our problem set-up is illustrated in Fig. 1: The target phenomenon is largely unknown, and measurements are only available for the locations that were observed by the robot team (as opposed to the exteroceptive sensing that is discussed commonly in the robotics literature). We assume that each robot moves deterministically. Each robot models its current knowledge about the phenomenon using a GP. For the other unvisited locations, we use the ensemble of distributed GPs to make a prediction and then fuse predictions from all the experts in order to have a unified prediction. Each robot gathers its own observations (from the whole field) during its mission time and egotistically optimizes its GP model to make a model for the whole phenomenon. Given individual training sets for each expert, they are bound to have slightly conflicting predictions over unvisited locations. Hence, we fuse the predictions from all the experts to deduce an overall prediction model. Planning informative tours under resource constraints is equally applicable in other applications like efficient routing of a truck through *a priori* known depot locations under limited budget while maximizing the net reward accrued [23] or in monitoring oceanic oil spills, maximizing electoral turn over [22], and the like. Note that we use two-dimensional (2-D) spatial interpolation in this figure for ease of visualization. However, we do not intend to use this interpolated data for training and inference of GP models.

This paper addresses the following research problem: *Given multiple robots, each acting as a self-reliant GP expert, how can we effectively fuse predictions in order to make a globally consistent model of the target environment when communication channels are not available?*

III. Distributed Gaussian Process Model of the Target Phenomenon

In this section, we summarize our idea of modeling the environment using a fully decentralized team of mobile robots. For this, each robot behaves like a GP expert, which, in a fully decentralized fashion, models the target phenomenon based on its own observations. Together, the team behaves like a distributed ensemble of GP experts as follows: For a domain $D \subset \mathbb{R}^d$, we model the phenomenon $f : \mathbb{R}^d \to \mathbb{R}$ using a GP, such that $f \sim GP(\boldsymbol{\mu}(\cdot), k_f(\cdot,\cdot))$. For each input $x \in D$, we define the associated measurement by z_x if the input was observed, otherwise we define $Z_x = f(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ (where ϵ is independent identically distributed Gaussian measurement noise) as a random variable, which is used for predicting the measurements of unobserved inputs. Then, $\{Z_x\}_{x \in D}$ is a GP, such that each of its finite subsets is a multivariate Gaussian distribution [1].

A. Field Modeling Using the Gaussian Process

GPs are a rich class of nonparametric Bayesian models, which allow us to model spatial variations of the environmental phenomena. GPs consistently quantify the uncertainty associated with predictions, which can be exploited by active sensing schemes for exploration and obtaining the most informative sensing locations for each mobile robot.

A GP is a generalization of a Gaussian distribution and fully defined by a mean function $\boldsymbol{\mu}(\cdot) = \mathbb{E}[f(\cdot)]$ and covariance function $k_f(\cdot,\cdot)$. The covariance function (also known as kernel) defines the spatial correlation structure of the function to be modeled and is parameterized by a set of hyper-parameters denoted by $\boldsymbol{\theta}$. A commonly used covariance function is the squared exponential

$$\sigma_{xx'} = \sigma_{\text{sig}}^2 \exp\left(-\frac{1}{2}(x-x')^T L^{-1}(x-x')\right) + \sigma_\epsilon^2 \delta_{xx'} \quad (1)$$

where $x, x' \in D, L = \text{diag}(l_1^2, \ldots, l_d^2)$, and l_i are characteristic length scales, which determine the relevance of the corresponding input dimension for modeling the target phenomenon. σ_{sig} corresponds to the amplitude of the signal to be modeled, whereas σ_ϵ describes the magnitude of the noise. The hyper-parameters are $\boldsymbol{\theta} \triangleq \{\sigma_{\text{sig}}, \sigma_\epsilon, l_1, l_2, \ldots, l_d\}$. The hyper-parameters are trained using the standard procedure of evidence (type-II marginal likelihood) maximization [1]. Evidence maximization avoids overfitting by automatically trading off data fit to model complexity. Since there are a total of K GP experts, and each of these models is independently optimized, we will obtain a super-set of hyper-parameters $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K]$.

When a column vector $\mathbf{z}_{O_k}$ of realized measurements[2] becomes available for a set $O_k \subset D$ of inputs for the k^{th} GP expert, we exploit these measurements to train the GP and predict the measurements for a set $U_k \subseteq D$ of unobserved inputs [3], [18]. The corresponding GP posterior predictive distribution for each of the $k \in \{1, 2, \ldots, K\}$ GP experts is given by

$$\boldsymbol{\mu}_{U_k|O_k,\theta_k} \triangleq \boldsymbol{\theta}_{U_k} + \Sigma_{U_k O_k|\theta_k} \Sigma_{O_k O_k|\theta_k}^{-1} (\mathbf{z}_{O_k} - \boldsymbol{\theta}_{O_K}) \quad (2)$$

$$\Sigma_{U_k U_k|O_k,\boldsymbol{\theta}_k} \triangleq \Sigma_{U_k U_k|\boldsymbol{\theta}_k} - \Sigma_{U_k O_k|\boldsymbol{\theta}_k} \Sigma_{O_k O_k|\boldsymbol{\theta}_k}^{-1} \Sigma_{O_k U_k|\boldsymbol{\theta}_k} \quad (3)$$

where $\boldsymbol{\mu}_{U_k|O_k,\boldsymbol{\theta}_k}$ is a column vector of means of the predicted (posterior) measurements of the phenomenon modeled by the expert k and $\Sigma_{U_k U_k|O_k,\boldsymbol{\theta}_k}$ is the corresponding predictive posterior covariance matrix. Using the kernel trick, we deduce $\Sigma_{xx'} = k_f(x, x')$.

B. Resource-Constrained Decentralized Active Sensing (RC-DAS)

In order to gather the training samples for the GP model, a robot usually relies on active sensing techniques. In [19], we described the current state-of-the-art active sensing technique, which we referred to as fully decentralized active sensing (full-DAS). In this scheme, the robot visits the most uncertain (highest entropy) locations and only tries to optimize the model performance. In doing so, the robot is forced to travel far-off and incurs prohibitively large travel costs. Thus, in the same work, we proposed an active sensing approach that trades off the selection of the most informative (higher entropy) location and the closest location (lower distance), *i.e.,* uncertainty versus distance optimization. This new approach was called RC-DAS. The algorithm is summarized in Algorithm 1. This active sensing scheme was used to obtain trained GP experts, the predictions of which were fused to obtain the global map. In our prior work, we used the Euclidean distance between the current location and the next location but from here on, we shall use the Haversine[3] distance as explained in Definition 1.

Algorithm 1: RC-DAS (D, *B*).

$\{x\}_{i=1}^K \leftarrow x_i^{[1]}$; $\{z\}_{i=1}^K \leftarrow NULL$;
$\{O\}_{i=1}^K \leftarrow NULL$; $K = 4$;
for agent $i = 1, \ldots, K$ **do**
  While $B > 0$ **do**

    /***SENSE***/

    $z_{x_i} \leftarrow Sense(x_i)$ ▷ obtain measurement
    $\mathbf{z}_i \leftarrow [\mathbf{z}_i; z_{x_i}]$ ▷ store observation
    $O_i \leftarrow [O_i; x_i]$ ▷ store location

    /***PLAN***/

    $\theta_i \leftarrow MLE(\mathbf{z}_i, O_i)$ ▷ obtain hyper-parameters
    ▷ deduce most uncertain locations
    $O^*{}_i \leftarrow CalcUncertainNeighbors(D, x_i)$
    ▷ Compute predicted measurements
    $\mu^*{}_i, \Sigma^*{}_i \leftarrow CompPosterior(z_i, O_i, O^*{}_i, \boldsymbol{\theta}_i)$
    ▷ RC-DAS Objective Function
    $Obj \triangleq \left(\alpha\, \mathbb{H}[Z_{U_{w_k}}|\mathbf{z}_{O_{w_k}}] - (1-\alpha)\ln(D_{Hav}(x - x^*)\right)$
    ▷ optimal *Next Best Location*
    $x^* \leftarrow \arg\max_{x \in O^*{}_i}(Obj)$

    /***ACT***/

    ▷ pass target location to robot controller
    $x_i \leftarrow MoveToNextBestLoc(x^*)$
    $B \leftarrow B - (S + T)$ ▷ update remaining budget
  end while
end for

Definition 1: Given the geographic coordinates of two locations $L_1, L_2 \in \mathbb{R}^2$, we can define the Haversine distance (D_{Hav}) between these locations as [24]

$$D_{\text{Hav}}(L_1, L_2) \triangleq 2r \arctan\left(\sqrt{\frac{a}{1-a}}\right),$$

$$a \triangleq \sin^2\left(\frac{y_2 - y_1}{2}\right) + \cos(y_1)\cos(y_2)\sin^2\left(\frac{x_2 - x_1}{2}\right)$$

where $r = 6371\,\text{km}$ represents the radius of the earth and $L_1 = (x_1, y_1); L_2 = (x_2, y_2)$ represent the respective latitudes and longitudes of the locations. Hence, the Haversine distance returns the separation between two locations while accounting for the curvature of the earth.

In Algorithm 1, we defined our RC-DAS algorithm where we tradeoff resource utilization to model performance in order to make an accurate model of a large-scale environmental phenomenon. The inputs to this algorithm are D, which is the domain of the algorithm, and B, which is the initially allocated budget for our robots. Usually for a robot, to move from its current location to the target location, three primitives are involved: sense, plan, and act. Based on [25] and in light of our problem setting, we define these primitives as follows.

[2]Can be observed only by mobile robots at particular locations and time instances.

[3]Since our dataset spans across continental USA, the travel costs are best explained by Haversine distance instead of its Euclidean counterpart. For considerably smaller state-space, we can rely on Euclidean or Haversine distance interchangeably.

Sense: The robot gathers measurements from its environment.

Plan: The GP experts process the measurements and make a prediction for each location in U_{global}. Moreover, the GP experts evaluate the optimal *next-best-sensing-location*, which is defined by our objective function.

Act: The target location is passed to the robot controller allowing the robot to move to the desired location and obtain new measurements to update the model. Additionally, the residual resources are updated.

Now we give an in-depth explanation of the steps for each of these primitives. When the robot is at a certain location x_i and obtains a sensor measurement z_{x_i}, we store the observation (line 1) and input location (line 1) for inference (line 1). We then evaluate the most uncertain neighbors surrounding the current robot location, which are within accessible limits of the robot (line 1). We then compute the posterior prediction over these locations, as shown in line 1. To evaluate the most informative next-best-sensing-location, we evaluate our proposed cost function to optimize the travel distance and simultaneously reduce the prediction uncertainty (line 1). If we jointly maximize over this cost function, we get the feasible next best location, as shown in line 1. We pass this as the current goal position to be attained by our mobile robot. In line 1, we update our remaining budget B available to the robot by subtracting the sensing cost (S) and travel cost (T) incurred by our agent when it moved to the new location x^*. Besides the factors considered in our cost function, there are other contributors like external wind effect and obstacle avoidance, which might lead to increased traveling costs but have not been considered in the scope of this work. Also, robot dynamics like rotor downwash, etc., which might influence and disturb the environment are beyond the scope of this work and have been studied elsewhere like [26] and [27].

IV. Predictive Model Fusion for Distributed GP Experts (FuDGE)

At the end of exploration (mission time) of all members of the mobile robot team, we obtain K diverse GP experts, which were each trained on their respective subsets of training data and have made a predictive map over the entire target phenomenon. To fuse the predictions from multiple models, we do a *consistency check*, which involves finding the probe (test) locations that are shared by all the GP experts.

Thus, we define $U_{\text{global}} \triangleq \{U_1 \cap U_2 \cap \cdots \cap U_K\}$ as the super set of all unobserved inputs that were never visited by any robot. Similarly, we define $O_{\text{global}} \triangleq \{O_1 \cup O_2 \cup \cdots \cup O_K\}$ as the super set of all observed inputs that were visited by all robots.

A. Fusion Strategy

We define a probe point $Q \in U_{\text{global}}$ as a point of interest for which we want to fuse the predictions from multiple GP experts and consider the following fusion algorithms for our analysis.

1) Pointwise Mixture of Experts Using the GMM: We first give the premise of our model for the ease of the readers followed by the elaborate model description.

Premise: GPs are kernel-based methods and since we use isotropic squared exponential kernels, as shown in (1), predictions are made with high confidence nearby the observed inputs and the confidence drops gradually as the distance increases [1]. This is also supported by the Tobler's first law of geography, which states that "Everything is related to everything else, but near things are more related than distant things" [28].

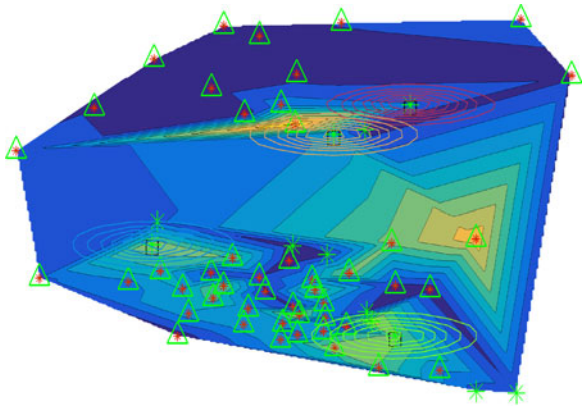

Fig. 2. FuDGE. Illustration of weighted fusion performed using *FuDGE* by positioning a 2-D Gaussian distribution $\mathcal{N}(x_i, \Sigma_k)$ to evaluate the responsibility of a GP expert over a probe point. In this figure, locations marked in green asterisk (*) represent the training locations that were visited by our robots during their respective missions, while those highlighted by red asterisk (*) represent the probe points over which we need to fuse predictions, and black squares (□) represent the start location of each of the four robots. For simplicity, we have only shown the first training sample of each of the GP expert. We have four experts, each of which is represented by a Gaussian contour plot centered around their first training sample.

Model description: We now explain our novel model fusion technique. For this, we obtain independently[4] trained GP experts from our distributed GP framework [19]. Then during the test phase, we combine the expert predictions based on the proximity of a test (probe) point to the experts' training samples. Thus, on the lower level, we develop independent prediction models, and on the higher level, we develop a fused globally consistent model making our model a two-level architecture.

The length scales inferred by the GP experts represent the standard deviation in spatial variation of measurements along the i^{th} input dimension σ_i. A probe point Q lying too far[5] away from the training points of the kth expert will not be confidently predicted by that GP. This is attributed to the fact that we use a stationary squared exponential covariance (1) to model our environment. Using this covariance structure, we can infer that correlation in measurements at two locations x and x^* will decay as the spatial separation between them increases. Thus, we place a multivariate Gaussian distribution over $O_k \sim \mathcal{N}\left(Q|O_k^j, \Sigma_k\right)$, where j represents the j^{th} training sample of the kth expert and $\Sigma_k \triangleq \text{diag}(l_{\text{lat}}^2, l_{\text{long}}^2)$. The spread of the multivariate normal distribution is defined in terms of length scales along the $Latitude$ and $Longitude$ of the corresponding GP expert. This gives rise to one Gaussian mixture model (GMM) over the training data points of each GP expert. We can define the responsibilities of this hierarchical GMM as

$$\log p(k|Q, O_{\text{global}}) \triangleq \Sigma_{x_i} \log p(Q|x_i, \Sigma_k). \tag{4}$$

In (4), x_i refers to $[O_{\text{global}}]_i$, Σ_k refers to the covariance of the Gaussian distribution for the kth GP expert, and $p(Q|x_i, \Sigma_k) = \mathcal{N}(x_i, \Sigma_k)$. This is illustrated in Fig. 2 where all the locations that were unvisited by the robot (k) during its exploration are referred to as the test set for that robot U_k. The responsibilities of a hierarchical GMM in (4) are such that $\log p(k|Q, O_{\text{global}}) \in [0, 1]$ and $\Sigma_{k=1}^K \log p(k|Q, O_{\text{global}}) = 1$.

[4]Not the same as conditional independence. Just refers to individual models maintained by each expert.

[5]Outside the 99.5% confidence bound.

We define the fused prediction at probe point Q as

$$\mu_{Q|O_k,\boldsymbol{\theta}_k} \triangleq \Sigma_{k=1}^{K}\left(\log p(k|Q,O_{\text{global}})\mu_k^Q\right). \tag{5}$$

In (5), the fused prediction at probe point Q is defined as the sum of predictions (μ_k^Q) weighted by the sum of log-responsibilities of a GMM $(\log p(k|Q,O_{\text{global}}))$ for each expert $k \in 1,\ldots,K$.

We now define the net variance at the probe point Q as

$$\begin{aligned}\sigma_{Q|O_k,\boldsymbol{\theta}_k} \triangleq\ & \Sigma_{k=1}^{K}\left\{\log p(k|Q,O_{\text{global}})[(\sigma_k^Q)^2+(\mu_k^Q)^2]\right\}\\ & -(\mu_{Q|O_k,\boldsymbol{\theta}_k})^2,\\ =\ & \Sigma_{k=1}^{K}\left(\log p(k|Q,O_{\text{global}})(\sigma_k^Q)^2\right)\\ & +\Sigma_{k=1}^{K}\left(\log p(k|Q,O_{\text{global}})(\mu_k^Q)^2\right)\\ & -(\mu_{Q|O_k,\boldsymbol{\theta}_k})^2. \end{aligned}\tag{6}$$

Since $(\cdot)^2$ is a convex operator, using Jensen's inequality [29], we know that $\Sigma_{k=1}^{K}\left(\log p(k|Q,O_{\text{global}})(\mu_k^Q)^2\right) \geq (\mu_{Q|O_k,\boldsymbol{\theta}_k})^2$. Now, (6) can be interpreted as the weighted combination of variances of the components plus a correction term, which is always positive. The correction term accounts for the divergence of respective component means (μ_k^Q) from the mean of the mixture $(\mu_{Q|O_k,\boldsymbol{\theta}_k})$ for the probe point Q.

2) Generalized Product-of-Experts Model [9]: This ensemble predicts the value at a test point as a weighted product of all expert predictions for this test point. The gPoE model allows us to flexibly define the weights to adjust for importance of an expert [2]. In the original work, a differential entropy score was used to define the weight of the experts based on the improvement in information gain between the prior and the posterior. Thus, in our setup, we define the weights (β_k) of the k^{th} expert, and fused predictions generated by an ensemble of K GP experts are obtained as follows:

$$\beta_k = \frac{1}{2}(\log(\sigma_{k_{**}}^2) - \log(\sigma_k^2(x^*))) \tag{7}$$

$$\hat{\beta}_k = \frac{\beta_k}{\Sigma_k \beta_k} \tag{8}$$

$$\mu_{U|O,\Theta}^{\text{gPoE}} \triangleq \Sigma_{UU|O,\theta_k}^{\text{gPoE}} \sum_{k=1}^{K} \hat{\beta}_k \Sigma_{UU|O_k,\boldsymbol{\theta}_k}^{-1} \mu_{U|O_k} \tag{9}$$

$$(\Sigma_{UU|O,\Theta}^{\text{gPoE}})^{-1} \triangleq \sum_{k=1}^{K} \hat{\beta}_k \Sigma_{UU|O_k,\boldsymbol{\theta}_k}^{-1}. \tag{10}$$

In (7), we define the differential entropy score, and in (8), we evaluate the confidence weight per probe point x^* by finding the differential entropy between the prior variance $\sigma_{k_{**}}^2$ and posterior variance $\sigma_k^2(x^*)$ for the probe point x^* such that $\Sigma_k \hat{\beta}_k = 1$.

This model, currently the state-of-the-art, is too conservative and ends up overestimating the variance [2]. This is owing to the fact that there is no variance correction term in (10). This is elegantly overcome by FuDGE.

V. Experiments

In this section, we compare the performance of individual robot prediction models with that of the fused models in order to empirically show that our data fusion, in fact, leads to enhanced prediction [8]. In order to evaluate the fusion performance, we compare our proposed model *FuDGE* to gPoE [9]. For all experiments, each robot was given access to all the locations over the whole field. We arbitrarily assigned the robot starting locations and averaged the performance over 5 experiments for each chosen length of walk (size of training samples per robot). We also analyzed the fusion performance with regards to increasing the length of walk, *i.e.,* increasing the number of training samples per robot. Note that each robot was only allowed to plan a tour through the locations that were provided in the dataset.

A. Dataset for Performance Evaluation

We used the *US Ozone Dataset*. This dataset includes ozone concentrations (in parts per billion) collected by US Environmental Protection Agency [19]. In this dataset, the measurements were recorded from 1995 to 2011 for 59 static monitoring stations across USA but we only choose one of the years for evaluation purposes. For each station, the annual average ozone concentration at a station was assigned as the sample measurement for that station. Since the environment to be monitored was quite large and the robot needs to cover distances of the order of a few hundred kilometers, we simulated high-end commercial drones like the one shown in Fig. 1, which can fly at significantly high speeds during which the correlations remain rather static. Also, as opposed to conventional batch processing approaches, we do not split the entire dataset into test and train subsets. Rather we allow the robots to utilize the RC-DAS selection mechanism to actively select the samples, which they deem necessary to be observed. Thus, as explained earlier on, the input locations that were observed by a robot during its mission time can be considered as its training set and the unobserved input locations can be considered as its test set.

B. Comparative Analysis of RC-DAS and Full-DAS

Here, we compare the two active sensing approaches based on their fusion performance and average path cost incurred.

1) Prediction Performance Analysis: In this section, we evaluate the average RMSE, which represents the average of errors of all robots between the estimated model and the ground truth evaluated over each element of U_{global}.

Fig. 3(a) shows the fusion performance of full-DAS against the average performance of independent robots labeled as IndepGP, the state-of-the-art gPoE and the single GP case evaluated over U_{global}. *IndepGP* refers to the average of individual performances of all robots as evaluated over U_{global}. As explained earlier on, this does not mean that our GPs are conditionally independent of each other, since they might have had shared training samples owing to uncoordinated exploration but we refer to the independence in the sense of uncoordinated individual GP expert models. We can see that the independent robots tend to incur a higher performance error (average) owing to limited exploration. This error tends to go down as more observations become available. However, fusion strategies outperform the independent robot models. By comparing Fig. 3(a) and (b), we can clearly see that full-DAS tends to perform better than RC-DAS since in this case, the GPs had access to the most uncertain and hence the most informative training samples. This also helps our fusion model perform better as some of the experts tend to know slightly more information about a region as compared to the others, and hence not all experts can be given equal weights. From Fig. 3(a) and (b), it can be observed that the average fusion performance of our proposed model is always the best. In essence, our FuDGE can be considered a simple averaging when the GP experts are equally good (or bad) at predictions for a probe point while at other times, FuDGE assigns the weights to GP experts based on the log-likelihood of the GP for the probe point. Another interesting fact to note here is that, while the error of all fusion techniques for

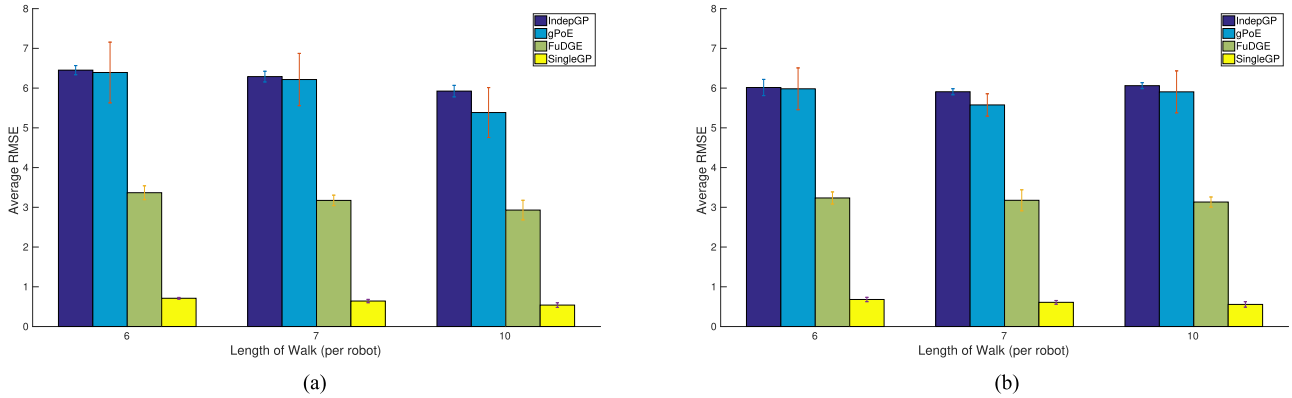

Fig. 3. Fusion performance. Evaluating average fusion performance for full-DAS and RC-DAS versus length of walk [Ozone dataset]. (a) Performance of FuDGE using full-DAS. (b) Performance of FuDGE using RC-DAS.

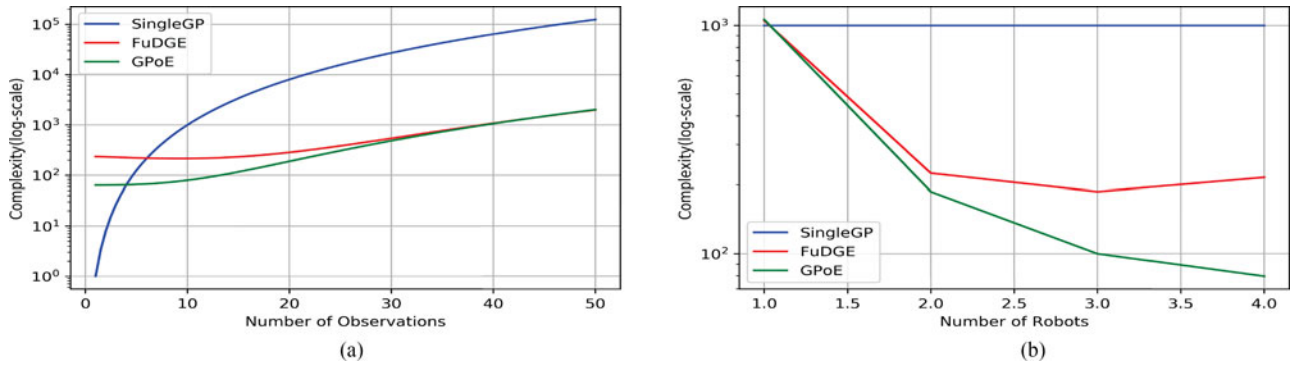

Fig. 4. Computational complexity. Illustrating the computational complexity of singleGP, FuDGE, and gPoE models. (a) Complexity versus length of walk. (b) Complexity versus number of robots.

full-DAS tends to reduce as we increase the length of walk of each robot, the error does not follow a monotonically decreasing trend for RC-DAS owing to the choice of training samples as explained earlier. Moreover, since FuDGE and gPoE are approximations of a single GP to aid efficient robot exploration, we can see that this is achieved at the cost of compromised accuracy.

In order to check the statistical significance of our claims, we deduce the *p-values* for our experiments. For this we define our null hypothesis H_0:FuDGE does not perform better than gPoE and our alternative hypothesis H_a: FuDGE performs better than gPoE. Then for full-DAS and RC-DAS, we evaluate the *p-values* of the *z-statistic* for the right-tailed test as 0.0294 and 0.0090, respectively. We choose a significance level of $\alpha = 0.05$ and since $p < \alpha$ for both active sensing techniques, we have strong evidence against the null hypothesis, so we reject the null hypothesis. Thus, the performance of FuDGE is significantly better when compared to that of gPoE.

2) Path Cost Analysis: We also evaluated the path cost incurred by both active sensing schemes, whereby the path cost refers to the net sensing cost and traveling cost incurred by a robot during its exploration. The results are summarized in Fig. 5, which shows the average past cost representing the average of total path costs incurred by all robots. It can be seen that the costs incurred by full-DAS are consistently higher than that of RC-DAS. This goes to satisfy our claims that we can successfully tradeoff model performance to efficient resource utilization without having to drastically compromise on any one of them.

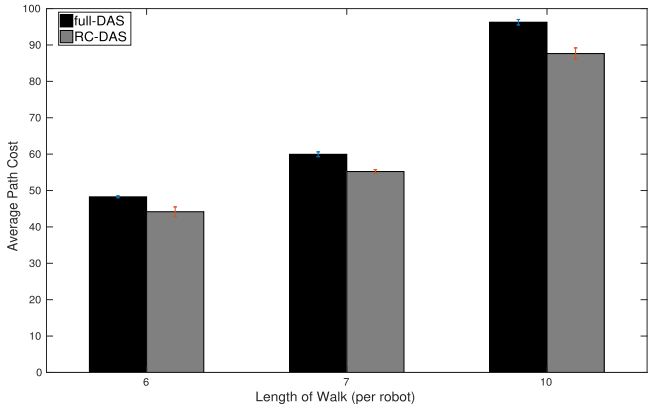

Fig. 5. Path cost. Evaluating path cost for full-DAS and RC-DAS versus length of walk [Ozone dataset].

In conclusion, it is apparent that both full-DAS and RC-DAS attain similar performance in terms of fusion quality but RC-DAS does so at lower path costs.

C. Model Complexity Analysis

From Fig. 3, it is explicit that FuDGE outperforms existing state-of-the-art models but it is also essential to analyze that the performance

TABLE I
COMPUTATIONAL COMPLEXITY ANALYSIS FOR FUDGE, GPOE, AND SINGLEGP

	Inference complexity	Exploration complexity	Fusion complexity	Instances
SingleGP	$I_S = \mathcal{O}(S)^3$	$E_S = \mathcal{O}(R_S)$	$F_S = \emptyset$	$S = \#(O_{\text{global}}), R_S = \#(D \setminus O_{\text{global}})$
FuDGE	$I_F = \mathcal{O}(\frac{S}{K})^3$	$E_F = \mathcal{O}(R_F K)$	$F_F = \mathcal{O}(FK)$	$F = \#(U_{\text{global}}), R_F = \#(D \setminus O_k)$
gPoE	$I_G = \mathcal{O}(\frac{S}{K})^3$	$E_G = \mathcal{O}(R_G K)$	$F_G = \mathcal{O}(G + K)$	$G = \#(D), R_G = \#(D \setminus O_k)$

was not obtained at the cost of extensive computations. In order to perform a fair comparison between the referred models, we obtain the trajectories of all robots *a priori* using RC-DAS. Then, we feed the same trajectories to all models and account for this cost of exploration as if the model was performing exploration in real time. For this, we define $\#(\cdot)$ as the cardinality operator and K represents the size of the team operating in the field whose domain is D as before. The results, hence, obtained are summarized in Table I.

In Table I, we categorize the computational cost using three components: all costs referenced with I_* refer to the cost for performing GP inference, E_* refer to the computational cost for active sensing, and F_* refer to the computational cost for fusion. As a visual representation of Table I, we also show Fig. 4 wherein we show the model complexities with growing number of observations for a fixed size of team [see Fig. 4(a)] and also the impact of variable size of team [see Fig. 4(b)]. From Fig. 4(a), it is clear that both FuDGE and gPoE are computationally lighter than singleGP, and as the number of observations grows, FuDGE and gPoE are computationally equivalent but FuDGE is more accurate. In Fig. 4(b), we show that FuDGE and gPoE are better off than their SingleGP counterpart as they can efficiently distribute the computational load over the entire fleet. Thus, based on Fig. 4, we can arrange the three models in decreasing order of complexity as: $SingleGP > FuDGE \geq GPoE$. From this, we conclude that not only FuDGE can generate significantly better fused maps as compared to existing state-of-the-art models, but this is also done at equivalent or nominally higher computation costs.

VI. CONCLUSION AND FUTURE WORK

In this paper, we introduced FuDGE, for point-wise fusion of predictions and resolving conflicting models when multiple robots try to model a spatially varying environment based on limited observations. Our empirical results show that we can elegantly tradeoff model performance to resource utilization without drastically compromising on either. This work is just a preliminary step and opens up new areas of exploration.

Further enhancements could include but are not limited to: First, development of partial decentralization schemes like the works of [3] so that the robots can coordinate the paths with its nearest neighbors to avoid gathering overlapping observations. Second, we could replace our kernel function with area kernels [30] to consider continuous measurements, which when coupled with [31] can help us plan trajectories over multiple time steps to infer both spatial and temporal dynamics of the environment. Also, in doing so, since we can utilize a fully Bayesian inference, we believe that the experts can soon infer the locally stationary clusters of a fully nonstationary field, which may be assigned to each expert like [18]. Perhaps monitoring only the locally stationary clusters could be more effective than our current approach so we would further investigate this. Third, a more complex stacked model can be investigated based on [32], such that we can fuse multiple GP experts into a single GP expert, which outperforms each individual expert. Finally, our model should be made fault-tolerant by taking into account erroneous local models that might have to be used for fusion.

Furthermore, we have only looked into informative path-planning in discrete scenarios like the works of [22] where the locations are predefined and measurements are only available at these locations. However, it might be interesting to investigate a continuous field and endow the robots with the freedom to fly from its current location to any other location within the field limits to observe and gather measurements. This could allow robots to gather dense samples of measurements and make the problem more close to real life scenarios. However, in doing so, we would incur whole new computational complexities and optimization challenges, and hence has been left for further works.

ACKNOWLEDGMENT

The authors would like to thank Prof. M. P. Deisenroth for preliminary discussions that helped lay the foundations of their model. Additionally, the authors would like to thank the anonymous reviewers whose comments have helped to enhance the quality of this work.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[2] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, vol. 2, no. 2, 2015, p. 5.
[3] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 901–921, Jul. 2015.
[4] A. G. Wilson, C. Dann, and H. Nickisch, "Thoughts on massively scalable Gaussian processes," arXiv:1511.01870.
[5] S. Vasudevan, "Data fusion with Gaussian processes," *Robot. Auton. Syst.*, vol. 60, no. 12, pp. 1528–1544, 2012.
[6] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Heteroscedastic Gaussian processes for data fusion in large scale terrain modeling," in *Proc. 2010 IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3452–3459.
[7] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Non-Stationary dependent Gaussian processes for data fusion in large-scale terrain modeling," in *Proc. 2011 IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1875–1882.
[8] S. Vasudevan, A. Melkumyan, and S. Scheding, "Efficacy of data fusion using convolved multi-output Gaussian processes," *J. Data Sci.*, vol. 13, pp. 341–367, 2015.
[9] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of Gaussian process predictions," *Proc. Mod. Nonparametrics 3: Autom. Learn. Pipeline Workshop NIPS*, Montreal, QC, Canada, 2014.
[10] E. Meeds and S. Osindero, "An alternative infinite mixture of Gaussian process experts," *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 883–890, 2006.
[11] C. Yuan and C. Neubauer, "Variational mixture of Gaussian process experts," *Adv. Neural Inf. Process. Syst.*, vol. 21, pp. 1897–1904, 2009.
[12] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1177–1193, Aug. 2012.
[13] V. Tresp, "A bayesian committee machine," *Neural Comput.*, vol. 12, no. 11, pp. 2719–2741, 2000.
[14] A. Viseras, T. Wiedemann, C. Manss, and L. Magel, "decentralized multi-agent exploration with online-learning of Gaussian process," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 4222–4229.

[15] F. Xue, R. Subbu, and P. Bonissone, "Locally weighted fusion of multiple predictive models," in *Proc. Int. Joint Conf. Neural Netw.*, 2006, pp. 2137–2143.
[16] P. Baraldi, A. Cammi, F. Mangili, and E. E. Zio, "Local fusion of an ensemble of models for the reconstruction of faulty signals," *IEEE Trans. Nucl. Sci.*, vol. 57, no. 2, pp. 793–806, Apr. 2010.
[17] F. Lavancier and P. Rochet, "A general procedure to combine estimators," *Comput. Statist. Data Anal.*, vol. 94, pp. 175–192, 2016.
[18] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-robot active sensing of non-stationary Gaussian process-based environmental phenomena," in *Proc. 2014 Int. Conf. Auton. Agents Multi-Agent Syst.*, 2014, pp. 573–580.
[19] K. Tiwari, V. Honoré, S. Jeong, N. Y. Chong, and M. P. Deisenroth, "Resource-constrained decentralized active sensing for multi-robot systems using distributed Gaussian processes," in *Proc. 2016 16th Int. Conf. Control, Autom. Syst.*, 2016, pp. 13–18.
[20] Y. Xu and J. Choi, "Adaptive sampling for learning Gaussian processes using mobile sensor networks," *Sensors*, vol. 11, no. 3, pp. 3051–3066, 2011.
[21] K. Tiwari, S. Jeong, and N. Y. Chong, "Map-reduce Gaussian process (MR-GP) for multi-UAV based environment monitoring with limited battery," in *Proc. 2017 56th Annu. Conf. Soc. Instrum. Control Eng. Jpn.*, 2017, pp. 760–763.
[22] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1106–1118, Oct. 2016.
[23] Z. Ma, K. Yin, L. Liu, and G. S. Sukhatme, "A spatio-temporal representation for the orienteering problem with time-varying profits," *Proc. 2017 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 6785–6792, Sep. 2017, doi: 10.1109/IROS.2017.8206597.
[24] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Destination prediction by trajectory distribution-based model," *IEEE Trans. Intell. Transpot. Syst.*, to be published. doi: 10.1109/TITS.2017.2749413.
[25] R. Murphy, *An Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA, USA: Bradford, 2000.
[26] M. Reggente and A. J. Lilienthal, "Using local wind information for gas distribution mapping in outdoor environments with a mobile robot," in *Proc. 2000 IEEE Sens.*, Oct. 2009, pp. 1715–1720.
[27] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Autonomous gas-sensitive microdrone," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 50–61, Mar. 2012.
[28] N. Waters, "Tobler's first law of geography," in, *The International Encyclopedia of Geography*. Hoboken, NJ, USA: Wiley, 2017.
[29] M. Kuczma, *An Introduction to the Theory of Functional Equations and Inequalities: Cauchy's Equation and Jensen's Inequality*. New York, NY, USA: Springer, 2009.
[30] C. E. O. Vido and F. Ramos, "From grids to continuous occupancy maps through area kernels," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1043–1048.
[31] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Proc. 2014 IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6136–6143.
[32] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.