2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Distributed Robotic Sampling of Non-homogeneous Spatio-Temporal Fields via Recursive Geometric Sub-division

Young-Ho Kim and Dylan A. Shell

Abstract— **Environmental monitoring, an important application for robots, has begun to be addressed recently with linear least squares regression techniques because they estimate the values of measured attributes and their uncertainty. But several challenges remain when performing adaptive sampling in a communication-constrained distributed multi-robot setting. When the attributes of interest evolve over time (as is natural for many environments) any non-homogeneous spatial variability may necessitate continual re-modeling of the field dynamics and/or re-sampling of the field. This raises questions about the robots' division of labor and workload balance that can be difficult to address when sample information is not stored centrally. This paper tackles these coordination problems efficiently by introducing a sub-division-based modeling technique appropriate for distributed decision-making. We augment Ordinary Kriging to enable representation of a field's (potentially non-homogeneous) evolution through Bayes filtering that characterize the underlying dynamics. This approach not only enables adaptive path planning in the field, but the sub-divided areas lead to a straightforward formulation of the optimal workload distribution through modification of an approximate graph partitioning algorithm. Using a simulated multi-robot sampling scenario, we demonstrate and validate the approach. The experiments show good performance in terms of cross-validation using real values and illustrate how *hotspots* are identified and modeled, in turn affecting the division of labor.**

I. Introduction

Environmental monitoring with robots encompasses a wide range of important applications [1], [2]; for example, a large-scale lake monitoring effort may help to detect contamination and help ensure the safety of the water. Recent research has made significant progress in predicting, monitoring, and tracking large scalar fields; including experimental work in aquatic, terrestrial, and subsoil settings [3]–[5]. The underlying challenge addressed by such systems stems from the measured data being sparse compared to the large spatial areas/volumes of interest. Although these problems are a natural fit for multi-robot systems, models that can scale to large fields, while being sufficiently rich to capture aspects of temporal variability and avoiding the proliferation of tuning parameters, give inadequate attention to practical considerations needed for distributed planning and decision-making.

In the context of multi-robot adaptive sampling, this paper considers two problems: *(i)* how does one model a non-homogeneous time-varying field efficiently? *(ii)* how does one distribute the robots' workloads (since a changing field may require non-homogeneous sample densities)? The majority of research, including recent work (*e.g.,* [6]), focuses on the first problem in isolation. In this paper, we consider that an appropriate answer to first problem must be considered in terms of the second problem and its context. The two are directly related because complex, dynamic fields befit multiple robots, each making autonomous decisions in an online fashion.

This paper represents temporal dynamics as uncertainty that is accrued with time. The rate of growth of this uncertainty is estimated, simplifying the covariance (or variogram) model, and reducing the complexity needed to treat the relationship between space and time. This is a pessimistic view born of the observation that treating data as *out of date* is usually more practical than fitting high-order non-linear models, especially given comparatively few samples. The method we propose incorporates a Bayesian filter to track this rate of growth of the field.

Non-homogeneity is captured by repeatedly dividing the field representation in a recursive manner, while maintaining the continuity of the field estimate (although, potentially sacrificing it in the uncertainty estimate). These sub-divided regions become units for assigning sampling tasks to each of the robots so as to balance the total workload. The division operation may cause the transfer of a region from one robot to another in its local vicinity; only summary information ever need be transferred between robots.

Specifically, the work contributes:

- An efficient recursive geometric sub-division of the field to balance and distribute the multi-robot workload.
- A simple unified Ordinary Kriging formulation that explicitly integrates a time-based relationship into the interpolator's uncertainty for fields undergoing temporal evolution.
- A novel probabilistic filtering to update the environmental model to estimate non-homogeneous field dynamics.

II. Related work

Linear least-squares estimation methods have been successfully employed in robotics for spatial interpolation and regional sampling. By way of recent examples, we mention work of Singh et al. [7] in the Gaussian Process Regression (GPR) framework, and Zhu et al. [8] via Kriging interpolation. Both use equivalent minimum error-variance estimation techniques that permit measured data to be interpolated in a way that takes into account a statistical description of a spatial variogram or covariance.

Employing these techniques to the model both spatial and temporal variability is a classical problem [9]. Singh et al. [7] used variations of these models with either stationary or non-stationary, and separable or non-separable spatial temporal covariance functions. However, these have many parameters, which are difficult to provide estimates (or even priors) for

Young-Ho Kim and Dylan A. Shell are with Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA. {yhkim, dshell}@cse.tamu.edu

and may need to adapt over time. Ultimately, the changing field is likely to be spatially non-homogeneous.

Several researchers are working on adaptive path planning in dynamic fields. Their solutions, however, involve a given (and fixed) covariance, augmented with other adhoc models. For example, Williams and Sukhatme [2] model the spatial dependency but add a plume-like process to generate a path that tracks the gradient. Recently, Smith N. et al. [1] employ an ocean phenomena forecast model.

Non-homogeneous temporal variability was recently tackled by [6], which is the closest work to the present paper. However, in contrast to the preceding work, our method decouples the spatial structure (captured via a purely spatial variogram) from temporal structure, where parameters model the latter change at a different time-scale to the former. This simplified representation facilitates efficient multi-robot communication. This is important because distributed multi-robot systems are a natural fit to adaptive sampling problems. We emphasize that our approach computes paths online on the basis of newly observed variability (in contrast to [10]). Also, the workload calculation only requires sharing of a local model of the neighborhood's dynamics. This reduces communication overhead compared with recent work in decentralized active sampling with a broadcast model [11].

III. Preliminaries & Proposed Estimator

In this section, we briefly describe Kriging [9] as a preliminary step, and then we present the proposed estimator for time varying scalar fields. Let $\chi \subseteq \mathbb{R}^2$ represent the spatial domain of interest, and let $T \subseteq \mathbb{R}^+$ represent time.

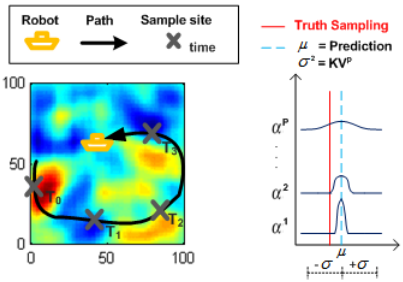
A. Ordinary kriging estimator

We use the Ordinary Kriging (OK) estimator [9] to make inferences on an unobserved value of the random process $Z^*(x_0)$, where x_0 is an unknown spatial location and $x_0 \in \chi$. The OK estimator makes use of samples, $Z(x_i)$, which are observed values at known spatial locations $x_i \in \chi$, $i \in [1, n]$, with a total of n observations. The estimate of $Z^*(x_0)$ is obtained from λ_i and n samples of $Z(x_i)$ with $\sum_i \lambda_i = 1$ as such [9]:

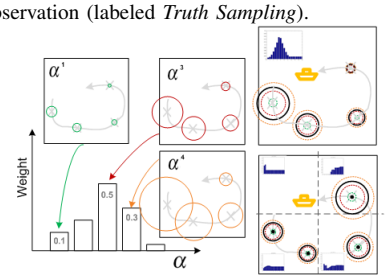
$$Z^*(x_0) = \sum_{i=1}^{n} \lambda_i Z(x_i), \quad \lambda = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{A} = \begin{pmatrix} \gamma(x_1,x_1)\cdots\gamma(x_1,x_n)1 \\ \vdots \quad \ddots \quad \vdots \quad \vdots \\ \gamma(x_n,x_1)\cdots\gamma(x_n,x_n)1 \\ 1 \quad \cdots \quad 1 \quad 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \gamma(x_1,x_0) \\ \vdots \\ \gamma(x_n,x_0) \\ 1 \end{pmatrix}. \tag{1}$$

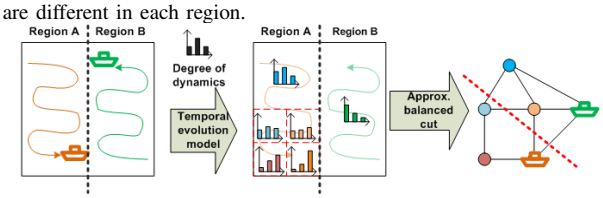
The spatio-temporal *semi-variogram* function γ captures the *strength* of spatial temporal relationships that represents the statistical effect of (stationary) relationships in space and time. However, the sorts of environmental attributes we wish to sample may evolve differently in different locations over time. Fixed variograms are inadequate to treat non-homogeneous temporal variability. The *Kriging Variance (KV)* $\sigma^2(x_0)$ represents the prediction uncertainty. $Z^*(x_0)$ and $\sigma^2(x_0)$ give the spatial prediction of unobserved values of the random process without considering either sensing noise or position uncertainty [9]. Our former work [5] shows



(a) The robot maintains several temporal models each of which is weighted and ultimately fused. *Left*: The robot traverses the field and makes measurements at different times. *Right*: A prediction from the field (μ with variance KV^p) is used to update weights via Bayes' rule and new observation (labeled *Truth Sampling*).



(b) Illustration of differently weighted models of temporal evolution. The increasing size of a circle over time shows the degree of temporal variability of each α^p, and how this is captured in the histogram of $\vec{\alpha}$, where the X-axis are scales α's and the Y-axis are weights ω's. Fields may be split or merged. The propagated rates (the histograms) are different in each region.



(c) Workload balancing in the multi-robot system via approximate graph partitioning. *Left*: Initially the robots have evenly separated regions. The field is updated as above in (a). *Middle*: Each robot constructs the associated sub-division graph. *Right*: An approximate algorithm facilitates the movement regions between robots to balance the total workload.

Fig. 1: **Overview of the proposed approach.** Robots represent temporal variability in the field as a form of uncertainty. See Sec. IV.

the explicitly separated sensing error from the variogram that was ignored before.

B. A measurement noise interpretation and estimation

To deal and simplify with spatial temporal fields, we model the passage of time as increasing error in the estimated values. A sample taken a long time ago is treated as if it involved significant measurement error by building on our earlier work in [5]. We add a function of time to the measurement error in order to represent the effect of changes in the field at the sample location that may have occurred since the measurement was made. The KV is used as an indication for the potential usefulness of sampling at a particular location. When time is incorporated in our proposed approach, it affords a natural way to include data staleness as part of that utility computation.

The simplicity of the method comes from the fact that Gaussian noise as a measurement error, is additive in the OK. The resultant variogram is $\gamma = \gamma^Y + (\sigma_i^2 + \sigma_j^2)/2$, where γ^Y is the basic variogram. Since most robotics work employs

formulations lacking such a treatment (*cf.* [7]), the derivation appears in the on-line supplemental material [12]. This shows a modified OK framework with different measurement errors.

Let Δt_i and Δt_j be the time elapsed since measuring at x_i and x_j, respectively, where x_i, $x_j \in \chi$. We suggest a propagated linear function of time, $N(0, \vec{\alpha}\Delta t_i)$ and $N(0, \vec{\alpha}\Delta t_j)$, where $\vec{\alpha}$ is scaling vector parameters. A set of values $(\vec{\alpha}, \vec{\omega})$ is $[(\alpha^1, ..., \alpha^P), (\omega^1, ..., \omega^P)]$, $p \in [1, P]$, where α^p are temporal scale values, ω^p are the (normalized) weights, and P is the number of the bins. We estimate $\vec{\alpha}$ by on-line using a probabilistic method: a Bayesian filter [13] is maintained for each scale parameter. It is natural to visualize ω^p ($\sum_{p=[1,P]} \omega^p = 1$) as vertical lines in a histogram for each α^p. Simply looking at this histogram allows one to interpret the temporal behavior of a region: when comparatively large α^p has significant probability ω^p then the field is fluctuating significantly over time. Detailed derivations are in [12].

Let $\hat{G}(x, t)$ be a function that represents the algorithm's output such that $\hat{G} : (\vec{\alpha}, \vec{\omega}) \rightarrow \mathbb{R}$, where $\hat{G}$ is $\sum_{p=[1,P]} \alpha^p \omega^p$ for χ. Then, we can derive one integrated $A_{ij} = \sum_{p=[1,P]} \omega^p A^p_{ij}$ and $b_{i0} = \sum_{p=[1,P]} \omega^p b^p_{i0}$. We define the modified OK for the weighted temporal evolution model, which gives one integrated KV (denoted KV^*):

$$\begin{aligned} A_{ij} &= \gamma^Y(x_i, x_j) + \frac{1}{2}\hat{G}(\Delta t_i + \Delta t_j) \\ b_{i0} &= \gamma^Y(x_i, x_0) + \hat{G}\Delta t_i, \end{aligned} \tag{2}$$

where $A_{ij} \in \mathbf{A}$, $b_{i0} \in \mathbf{b}$, and $i, j \in [1, n]$.

This yields the expected value of the evolving field by aggregating each α^p distribution with weight ω^p, and $(\vec{\alpha}, \vec{\omega})$ describes the scale of the temporal evolution, modelled as a linear change but with unknown drift.

IV. Distributed adaptive sampling

An overview of our approach is explained visually in Fig. 1. There are two important steps: the first step is to model the temporal evolution of the scalar field and to plan an adaptive path, as shown in Figs. 1(a) and 1(b) explained in Secs. IV-A and IV-B, respectively. the second step is to distribute the sampling to multiple robots and to balance workloads as illustrated in Fig. 1(c) explained in Sec. IV-C.

A. Estimation of adaptive field dynamics

This section presents an approach for estimating the temporal evolution in Fig. 1(a). We can compute a prediction $Z^*(x_0)$, and the variance KV^p with regard to each probabilistic model p. Then, the temporal variation of the field can be updated by $P(\omega^p_t | \omega^p_{t-1}, \tilde{\mathbf{Z}}_{\mathbf{t-1}}, Z_t)$, where $\tilde{\mathbf{Z}}_{\mathbf{t-1}}$ is a past observation set $[Z(x_1), ...Z(x_n)]$ and Z_t is the measurement $Z(x_0)$ at time t, and η is the normalization constant.

Line 4 of Alg. 1 represents a prediction step as a hypothetical probability distribution at time t based on $(\vec{\alpha}_{t-1}, \vec{\omega}_{t-1})$ is generated. This shows the update step and then Line 5 in Alg. 1 represents the Bayes filter posterior. Finally, we have a distribution, $(\vec{\alpha}_t, \vec{\omega}_t)$.

B. Adaptive path planning

Our path planning strategy consists of three parts: *(1)* separation of the regions, *(2)* generating queries (a set of potential solutions) in continuous space, and *(3)* maximizing a utility function to choose a solution. To better represent temporal variability, the region χ is subdivided by recursively decomposing the region into four equal quadrants, where $\bigcup_{j=1}^{4} \chi_{ij} = \chi_i$ (we simply assume four even sub-regions, which facilitate the implementation by a quad-tree). As we subdivide the regions recursively, we have updated K non-overlapping regions χ_i, where $i \in$ [1,K]. Therefore, $\bigcup_{i=1}^{K} \chi_i = \chi$, and $\chi_i \bigcap \chi_j = \emptyset$, $i \neq j$, and $i, j \in$ [1,K]. Each χ_i has the individual rate of temporal change $\hat{G}_i$. Let Z_t be a measurement at x_0 at time t, which updates a temporal evolution model in Section IV-A. Thereafter, if $x_0 \in \chi_i$, we can separate or merge regions based on the expected value of temporal change difference between $\hat{G}_{\chi_i}$ and $\hat{G}_{\chi_{i1}}$, $\hat{G}_{\chi_{i2}}$, $\hat{G}_{\chi_{i3}}$, $\hat{G}_{\chi_{i4}}$ where $\bigcup_{j=1}^{4} \chi_{ij} = \chi_i$. Based on this notation and definition, we find the maximum difference of the change rate of the spatial field and then subdivide the region so that our knowledge for each region is maximized. M_{cc} is the Maximizing Cross-Correlation (differentiation) between sub-regions:

$$M^i_{cc} \equiv max_{b=[1,4]}(\hat{G}_{\chi_i}, \hat{G}_{\chi_{ib}}), \quad \forall_i. \tag{3}$$

This determines that the χ_i region can be separated into four sub-regions χ_{i1}, χ_{i2}, χ_{i3}, and χ_{i4}, or maintained within a current region with regard to a new observation.

Next, we need the robot's next goal position x_0 at time t. First, We select location x_0 from a set of samples $[x_1, ..., x_Q]$, where Q is the number of potential locations. To generate the query set in the continuous space, the idea is to distribute the number of queries based on the importance of each region. Since $\hat{G}_i$ is the expected value of the rate of change in χ_i, we define the importance of the region χ_i to be proportional to the area size $|\chi_i| \times \hat{G}_i$. Randomly selected query locations are produced in subregions in Fig. 4. This implicitly captures the notion of *hotspot* sampling.

Finally, we pick the next goal position as the one that maximizes the utility function from the generated queries x_q, where $q \in [1, Q]$. Ideally, the utility represents how much each query minimizes $\int KV^*$, where r is the current robot position and x_q is an element of a query set. This is approximated as a discrete space, yielding:

$$\begin{aligned} U(r, x_q)_{q=[1,Q]} = \sum_{i=[1,Q], i \neq q} & (KV^*(r, x_i | (\vec{\alpha}, \vec{\omega})_{t-1}, \tilde{\mathbf{Z}}_{\mathbf{t-1}}) \\ & - KV^*(r, x_i | x_q, (\vec{\alpha}, \vec{\omega})_{t-1}, \tilde{\mathbf{Z}}_{\mathbf{t-1}})). \end{aligned} \tag{4}$$

Alg. 2 details adaptive path planning for this robot. Lines 4 to 12 in Alg. 2 separates or merges the fields by comparing M^k_{cc} and the threshold ϵ. The number of fields can be increased and decreased when we split or merge the

Algorithm 1 The Field Dynamics Filter

1: INPUT: at time t: field dynamics $(\tilde{\alpha}_{\mathbf{t-1}}, \vec{\omega}_{t-1})$, past observation set $\tilde{\mathbf{Z}}_{\mathbf{t-1}}$, and new measurement Z_t.
2: OUTPUT: $\omega^p_t, \forall p$.
3: **for** p = 1 to P **do**
4: (Z^*_t, KV^p_t) = modified OK$(\tilde{\mathbf{Z}}_{\mathbf{t-1}}, \alpha^p_{t-1}, \omega^p_{t-1})$
5: $\omega^p_t = \eta P(Z_t | Z^*_t, KV^p_t)\omega^p_{t-1}$
6: **end for**

Algorithm 2 The ADAPTIVE PATH PLANNER

1: INPUT: $(\vec{\alpha}_t, \vec{\omega}_t)_k, \forall k$, r, a split condition ϵ and a total number of queries Q.
2: OUTPUT: x_0 and χ'.
3: χ' = $\hat{G}'$ = NULL
4: **for** k = 1 to K **do**
5: **if** $M_{cc}^k > \epsilon$ **then**
6: χ' = χ' + $[\chi_{k1}, \chi_{k2}, \chi_{k3}, \chi_{k4}]$
7: $\hat{G}'$ = $\hat{G}'$ + $[\hat{G}_{\chi_{k1}}, \hat{G}_{\chi_{k2}}, \hat{G}_{\chi_{k3}}, \hat{G}_{\chi_{k4}}]$
8: **else**
9: χ' = χ' + $[\chi_k]$
10: $\hat{G}'$ = $\hat{G}'$ + $[\hat{G}_{\chi_k}]$
11: **end if**
12: **end for**
13: *factor*= $\sum_{k=1}^{K} |\chi'_k| \times \hat{G}'_k$
14: **for** k = 1 to K **do**
15: N_k = $Q \times |\chi'_k| \times \hat{G}'_k \div$ *factor*
16: **end for**
17: Generate random queries based on N_k at $\chi'_k, \forall k$
18: **for** q = 1 to Q **do**
19: $U_q = U(r, x_q)$
20: **end for**
21: **return** x_0 and χ'

fields. Lines 13 through 17 in Alg. 2 allocate the number of queries in each field based on the importance of each field. In line 19 of Alg. 2, we find a potential goal position that maximizes a utility function.

Our method picks a single point rather than generating a whole path. When we consider an optimal path, we need to compute all possible paths, and then evaluate each total KV^* to pick a unique path that (maximally) reduces the total uncertainty. However, it is beneficial to adjust and re-plan an on-line path when the field changes over time.

C. Distributed multi-robot sampling

Regions also have an associated graphical representation. A region χ that is decomposed into K non-overlapping regions $\chi_1, \cdots, \chi_k, \cdots, \chi_K$ is represented, along with the connectivity relationships, as a graph $G = (V, E)$, where a vertex $v_i \in V$ denotes the region χ_k, and an edge $e_{ij} = (v_i, v_j) \in E$ connects a pair of vertices v_i and v_j when the associated regions are adjacent. Separate weights are associated with both the vertices and edges.

With m robots, we wish to balance the workload by the partitioning the field χ. This is achieved by splitting the graph G into m sub-graphs $G_i = (V, E)$ such that $\bigcup_{i=1}^{m} G_i = G$. We expect $K \geq m$, so each robot is associated with at least one region. Although m can be any number, for clarity of presentation, consider the two robot case, with robot r_i and r_j. Let w_v^i and w_e^i denote the weights of v and e of the sub-graph G_i, respectively. Assuming G_i has L number of χ and N number of edges, each $w_{v_a}^i$ computed by $|\chi_a| \times \hat{G}_a$ is the importance of the region, $a \in [1, L]$. $w_{e_b}^i$ is the distance between v_{a_1} and v_{a_2}, $b \in [1, N]$ and $a_1, a_2 \in [1, L]$. Then this problem can be formulated as the following optimization problem:

$$\arg \min_{(G_i, G_j)} \; [\max_{i \in [1,m]} (\sum_{\forall a,b} (w_{v_a}^i + w_{e_b}^i)) - \min_{j \in [1,m]} (\sum_{\forall c,d} (w_{v_c}^j + w_{e_d}^j))]. \tag{5}$$

When G_i of robot r_i has the maximum total cost and G_j of robot r_j has the minimum total cost, $i, j \in [1, m]$, our goal is to minimize the difference of the total cost between G_i and G_j, as illustrated in Fig. 1(c).

Since the m graph partitioning problem known to be NP-hard [14], we adopt the heuristic of [15] because of its good performance in adjusting the unbalanced partitions as well as its practical running speed $O(n^3)$. A modification is necessary. We alter the formula by considering the neighborhood cost only to move a vertex (a region) to the other robot's sub-graph. The m robots initially split the regions evenly between themselves. Each robot follows the steps in Section IV-B, then updates its model of the temporal dynamics. The robots communicate data about nearby regions like the size of the region and the information describing the local field dynamics in $(\vec{\alpha}_t, \vec{\omega}_t)$. Then the total cost associated with G_i is $\sum w_v^i + \sum w_e^i$, where v^i and e^i are subsets of V_i and E_i, respectively.

Since the goal is to minimize the difference between the maximum cost of G_i and the minimum cost of G_j, we move $\{v_i\}$ to G_j. So, $G_i^{'} = G_i - \{v_i\}$ and $G_j^{'} = G_j + \{v_i\}$ where $G_i^{'}$ and $G_j^{'}$ are new G_i and G_j, respectively. Suppose R_i represents the total allocated cost of robot i and S_{ij} is the set of neighborhood nodes between G_i and G_j. Alg. 3 includes further detail.

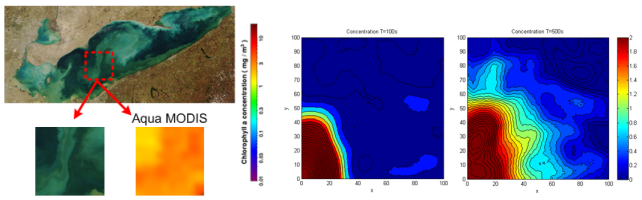
Algorithm 3 The DISTRIBUTED SAMPLING Algorithm

1: INPUT: sub-graphs G_m with M robots where $m = 1, ..., M$, current robot positions r_m.
2: OUTPUT: reconstructed sub-graphs G_m.
3: Compute $cost_{opt}$ = $\sum cost(G_m)/M$
4: **for** i = 1 to m **do**
5: Compute R_i = $\sum cost(G_i)$
6: **end for**
7: Find G_i of maximum cost R_i and G_j of minimum cost R_j
8: $S_{ij} = \{(v_i, v_j) | v_i \in G_i, v_j \in G_j\}$
9: $cost_{temp} = \infty$
10: **for** i = 1 to $|S_{ij}|$ **do**
11: $R_i^{'} = cost(G_i - \{v_i\})$, $R_j^{'} = cost(G_j + \{v_i\})$
12: **if** $(R_i - R_j) > (R_{i'} - R_j^{'})$ and $(R_i^{'} - R_j^{'}) > 0$ **then**
13: **if** $(R_i^{'} - R_j^{'}) < cost_{temp}$ **then**
14: $a = i$, $cost_{temp}$ = $(R_i^{'} - R_j^{'})$
15: **end if**
16: **end if**
17: **end for**
18: $G_i^{'} = G_i - \{v_a\}$ and $G_j^{'} = G_j + \{v_a\}$

V. SIMULATION AND EXPERIMENTS

A. Experimental setup

Using simulated data sets of chlorophyll gathered in the central basin of Lake Erie that is susceptible to severe oxygen depletion each summer [16], we verify the proposed methods to predict the proper chlorophyll levels. In practice, such data sets are spatially and temporally sparse. For example, NASA's Moderate Resolution Imaging Spectroradiometer (MODIS [17]) gathers data daily or hourly using satellite images with relatively low resolution that are affected by weather conditions. Data from the National Oceanic and Atmospheric Administration (NOAA) [16] is collected by fixed buoys which are spread across large areas (*e.g.,* 3 buoys per 100 km^2).



(a) Satellite and Aqua MODIS. (b) Simulated data at 100^{th} and 500^{th}

Fig. 2: This shows the plume-like ground truth data. 2(a) shows a real satellite image and MODIS image of central basin of Lake Erie in March 2012. The real satellite image of chlorophyll translated to the real rate of chlorophyll, and then Gaussian plume model is added into given sparse real data shown in 2(b).

To successfully run the experiments, we need to use dense data. Thus, we interpolate chlorophyll data for Lake Erie using the daily MODIS [17] data set (June to August 2012) and we opted to supplement the base field with a contaminant plume. A Gaussian plume model using two axes of diffusivity and velocity produces dynamics similar to what one might see in the ocean or a lake [18]. Finally, we have a interpolated dense data for 1000 steps over an 100 m^2 area to serve as the ground truth in Fig. 2.

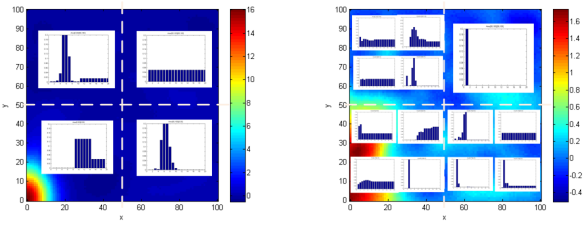
Two separate experimental scenarios were used, both of which assess the estimated variance in scalar fields constructed from measurements taken by robots:

1. Path planning and updating the temporal evolution model: The robot initially samples 20 randomly selected field values to construct the variogram, $\gamma^Y(h)$. The robot follows Alg. 2 to minimize temporal uncertainty over time and then updates the evolution model. We assume that the robot's velocity is constant, approximately $5-7\,m/s$. The greedy planner, which is selecting the next goal position in order to minimize the total uncertainty without considering the temporal changes [5], is used to compare the quality of our approach.

2. Distributed workload for multi-robot system: Each robot has an even initial partition to collect measurements and follows the above procedure. When the robots are collecting data following Alg. 2, Alg. 3 constructs and manipulates the graph partitions and distributes the workloads as observations are updated.

B. Results

Fig. 2 shows the models of field dynamics involving different time and space. Fig. 3(a) demonstrates four regions that have different field dynamics after the 69^{th} step. Fig. 3(b) represents 13 different field dynamics after the 217^{th} step. The field is separated resulting from Alg. 2. Some distributions of $(\vec{\alpha}_t, \vec{\omega}_t)$ are similar to the uniform distribution because when the region is separated, the field dynamics initially are computed from the uniform distribution if there are no past observations. It takes time to converge even if it already has several observations. Fig. 4 shows the steps for updating field dynamics over time. If the field has sufficiently large difference between values describing sub-regions and the current region, the current region is split or merged. Based on the differing field dynamics, the queries are distributed in the whole region. Once a split occurs, each region's temporal evolution model represents the region better than before. This helps the robots sample more efficiently in the *hotspot* region because KV^* increases in the hotspot quickly over time.



(a) 4 separations at 69^{th} step (b) 13 separations at 217^{th} step

Fig. 3: A robot estimates and updates the temporal evolution model with adaptive path planning. Each region has an original field dynamics model. Each histogram represents how much of a *hotspot* is formed. The X-axis is α^p and the Y-axis is ω^p.

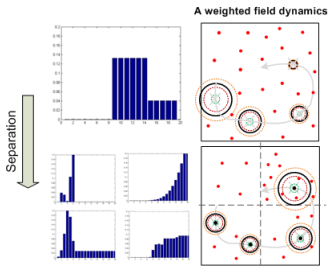
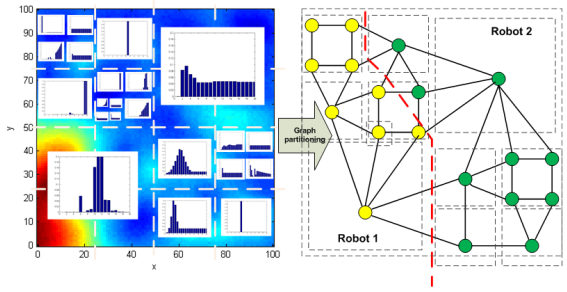


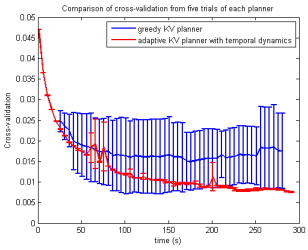
Fig. 4: The initial temporal evolution weight set is an uniform distribution, and then updated when the robot samples new data. If E_χ is enough to split or merge, each region has different ω distribution depends on the given data. X-axis is α^p, and Y-axis is ω^p. The red dots are the tentative queries. Each region has a different number of queries generated by the importance of temporal dynamics. Two cases shows a different degree of temporal variability depends on the histrogram of the region.

To evaluate the quality of our results, we show the cross-validation by Mean Square Error between ground truth and our estimation over time. We tested these for five trials shown in Fig. 5(b) which yields better cross-validation than the greedy planner. As we follow the scenarios of multi-robot case, our approach shows better cross-validations with low variance and faster convergence time.

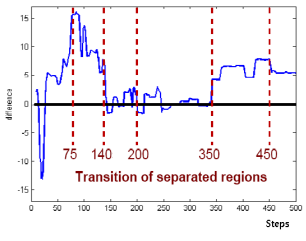
Fig. 5(a) demonstrates the distributed workloads for two robots. In Fig. 5(c), after a short amount of time, the difference of workload gradually approaches zero, showing that the workloads are well distributed. However, when a field is first split or merged, the workloads may be unbalanced for some time as the field model needs to be updated and to converge after the transition occurs. These transitions influence each robot's workload in the field as the robot samples new data. If each workload increases or decreases drastically, the overall workload can fluctuate until it eventually settles down with the split field. After the 350^{th} step, the workload increases for a while with the field redistribution by Alg. 3, but when another transition happens at the 450^{th} step, the workload gradually decreases. In other words, This shows non-monotonic after split transition and field redistribution. We believe that this settling time is related to the initial histogram distribution and redistributed split field.



(a) Distributed workload for two robots at 454^{th} step



(b) Cross-validations with ground truth



(c) Overall workload distribution over time

Fig. 5: The workload is distributed to two robots while each robot estimates and updates their field dynamics. 5(a) shows that the graph represents the workload including field dynamics, the size of each region, and the distance between nearby regions. 5(b) shows two planners that are cross-validations with ground truth. One is based on our approach with the ground truth over time. Another is the traditional approach with the ground truth over time. Data are from five separate trials of each case. 5(c) shows several sharp increases when the transition of separation is happened in the region (*e.g.*, 75^{th}, 140^{th}, 200^{th}, 350^{th}, and 450^{th} steps).

VI. Conclusion

This paper shows *(1)* how to model the temporal evolution field and *(2)* how to distribute the robots' workloads. The main idea is to treat the temporal variability as uncertainty in the interpolator. This approach can be seen as major simplification of a traditional spatio-temporal interpolation [9], which although widely known has seen comparatively little use in robotic sampling. The approach we outline facilitates modeling non-homogeneous temporal evolution as well as balancing the workload of the multi-robot system with efficient communication overhead, thus each robot adaptively has a balanced workload depending on the temporal variability of the assigned field.

The experiments cross-validate the estimated field with ground-truth values from the simulated data. The results show generally good performance even though the data have temporal variability. Moreover, in multi-robot experiments, the robots have a balanced workload, sharing only field dynamics and not the measurements themselves.

It is worth noting that the uncertainty estimates in this work are used predominantly for determining when a location should be sampled and then which robot should be responsible for carrying out the measurement. The simple — perhaps even simplistic— model suffices to ensure that the distributed system responds online to changes in the field by building an evolving model itself. A shortcoming of the uncertainty model, however, is that for robots i and j, where $e_{ij} \in E$, the $KV^*_{\chi_i}$ and $KV^*_{\chi_j}$ may not be continuous at the shared boundary. This means, for example, that the quad-tree structure may introduce some artefacts along the division line. The method of [6] can address this, but it remains unclear how such an implementation might be distributed in communication constrained settings.

Acknowledgement

This work was supported in part by the National Science Foundation as part of Grant IIS–1302393

References

[1] R. N. Smith, J. Das, Y. Chao, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Cooperative Multi-AUV Tracking of Phytoplankton Blooms based on Ocean Model Predictions," in *Proceedings of the IEEE/MTS Oceans Concerence and Exhibition*, Sydney, Australia, May 2010, pp. 1–10.

[2] R. K. Williams and G. S. Sukhatme, "Probabilistic Spatial Mapping and Curve Tracking in Distributed Multi-Agent Systems," in *Proceedings of the International Conference on Robotics and Automation*, St. Paul, Minnesota, USA, May 2012.

[3] V. Kumar, D. Rus, and G. S. Sukhatme, "Networked Robots," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer-Verlag Heidelberg, 2008, ch. 41.

[4] K. H. Low, J. M. Dolan, and P. Khosla, "Adaptive Multi-Robot Wide-Area Exploration and Mapping," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, May 2008.

[5] Y.-H. Kim, D. A. Shell, C. Ho, and S. Saripalli, "Spatial Interpolation for Robotic Sampling: Uncertainty with two Models of Variance," in *Proceedings of International Symposium on Experimental Robotics*, Quebec, Canada, June 2012.

[6] S. Garg, A. Singh, and F. Ramos, "Learning non-stationary space-time models for environmental monitoring," in *Proceedings of AAAI conference on Artificial intelligence*, 2012, pp. 288–294.

[7] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and Decision Making in Spatio-Temporal Processes for Environmental Surveillance," in *Proceedings of the International Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 2010, pp. 5490–5497.

[8] X. Zhu, J. Yu, S. Ren, and X. Wang, "Near-optimal Collecting Data Strategy Based on Orinary Kriging Variance," in *Proceedings of OCEANS*, Sydney, NSW, Australia, May 2010, pp. 1–6.

[9] N. Cressie and C. K. Wikle, *Statistics for Spatio-Temporal Data*. New York, NY, USA: John Wiley and Sons, Ltd, 2011.

[10] S. L. Smith, M. Schewager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.

[11] K. H. Low, J. Chen, J. M. Dolan, S. Chien, and D. R. Thompson, "Decentralized Active Robotic Exploration and Mapping for Probabilistic Field Classification in Environmental Sensing," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*, Valencia, Spain, June 2012.

[12] Y.-H. Kim and D. Shell, "Efficient Modeling of Non-homogeneous Field Dynamics for Distributed Robotic Sampling," http://www.cs.tamu.edu/media/27578/2014-2-1.pdf, Computer Sceince and Engineering, Texas A&M University, Tech. Rep. 2014-2-1, Feb. 2014.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[14] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified np-complete graph problems," *Theoretical Computer Science*, vol. 1, pp. 237–267, 1976.

[15] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Systems Technical Journal*, vol. 49, pp. 291–307, 1970.

[16] National Oceanic and Atmospheric Administration. (2012) NOAA Data Set. [Online]. Available: http://www.glerl.noaa.gov/data/

[17] NASA's Moderate Resolution Imaging Spectroradiometer. (2012) MODIS Data Set. [Online]. Available: http://oceancolor.gsfc.nasa.gov/

[18] E. Holzbecher, *Environmental Modeling: using Matlab*. New York, NY, USA: Springer-Verlag Heidelberg, 2007.