Efficient Modeling of Non-homogeneous Field Dynamics for Distributed Robotic Sampling*

Young-Ho Kim and Dylan A. Shell †

February 7, 2014

Abstract

Environmental monitoring is an important application for robotics — e.g., estimating pH or T° in marine settings. Linear least squares regression techniques have received much attention recently because they estimate the values of measured attributes and their uncertainty. Although the estimate of uncertainty aids in intelligent selection of subsequent measurement sites, several challenges remain when performing adaptive sampling in a communication-constrained distributed multi-robot setting. In particular, 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 of parameters 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.

*This is an extended version of a paper, “Distributed Robotic Sampling of Non-homogeneous Spatio-Temporal Fields via Recursive Geometric Sub-division”, selected to appear in the IEEE International Conference on Robotics and Automation of 2014

†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
1 Introduction

Environmental monitoring with robots encompasses a wide range of important applications [Smith et al., 2010; Williams and Sukhatme, 2012]; 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 [Kim et al., 2012; Kumar et al., 2008; Low et al., 2008]. 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., [Garg et al., 2012]), 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 spatio-temporal 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 use of a probabilistic filter to update the environmental model to estimate non-homogeneous field dynamics.

2 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., 2010] in the Gaussian Process Regression (GPR) framework, and [Zhu et al., 2010] 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 [Cressie and Wikle, 2011]. Singh et al. [2010] 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 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 [2012] model the spatial dependency but add a plume-like process to generate a path that tracks the gradient. Recently, Smith et al. [2010] employ an ocean phenomena forecast model.

Non-homogeneous temporal variability was recently tackled by Garg et al. [2012], 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 Smith et al., 2012). 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 that employs a broadcast model Low et al. 2012.

3 Preliminaries & proposed estimator

In this section, we briefly describe Kriging Cressie and Wikle, 2011 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.

3.1 Ordinary kriging estimator

We use the Ordinary Kriging (OK) estimator Cressie and Wikle, 2011 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 \( \lambda_i \) and \( n \) samples of \( Z(x_i) \) with \( \sum_i \lambda_i = 1 \) as such Cressie and Wikle, 2011:

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

\[
A = \begin{pmatrix}
\gamma(x_1, x_1) & \cdots & \gamma(x_1, x_n) \\
\vdots & \ddots & \vdots \\
\gamma(x_n, x_1) & \cdots & \gamma(x_n, x_n)
\end{pmatrix}, \quad b = \begin{pmatrix}
\gamma(x_1, x_0) \\
\vdots \\
\gamma(x_n, x_0)
\end{pmatrix}.
\]

(1)

The Kriging weights represented by \( \lambda \) are obtained from a distance-weighted average of nearby measurements. The spatial semi-variogram func-
tion $\gamma$ captures the strength of spatial relationships in the random field,

$$\gamma(x_i, x_j) = \frac{1}{2}E[(Z(x_i) - Z(x_j))^2],$$

where $x_i, x_j \in \chi$.

The Kriging Variance (KV) represents the prediction uncertainty given by

$$\sigma^2(x_0) = \sum_{i=1}^{n} \lambda_i \gamma(x_i, x_o) + \psi(x_o),$$

where $\psi(x_o)$ is a Lagrange Multiplier which is used in the minimization of the $\sigma^2(x_0)$. Note that Eq. (1) to Eq. (3) give the spatial prediction of unobserved values of the random process without considering either measurement noise by imperfect sensing or position uncertainty [Cressie and Wikle, 2011]. Our former work [Kim et al., 2012] shows the explicitly separated sensing error from the variogram that was ignored before.

The classic formulation for spatio-temporal field estimation [Cressie and Wikle, 2011] defines the general spatio-temporal model as

$$\gamma((x_i, x_j), (t_i, t_j)) = \frac{1}{2}E[(Z(x_i, t_i) - Z(x_j, t_j))^2],$$

which represents the statistical effect of (stationary) relationships in space and time, where $x_i, x_j \in \chi$ and $t_i, t_j \in T$. However, the sorts of environmental attributes we wish to sample with robots (e.g., involving hot-spots) may evolve differently in different locations over time. Fixed variograms are inadequate to treat non-homogeneous temporal variability.

### 3.2 A spatio-temporal model with a straightforward measurement noise interpretation

To deal with spatial temporal fields, we model the passage of time as increasing error in the estimated values. Even if uncertainty is not always increased (e.g., diurnal changes of temperature in a lake), our goal is to build a simple model rather than more predictable but required more knowledge. A sample taken a long time ago is treated as if it involved significant measurement error by building on our earlier work in [Kim et al., 2012], which formalized the OK to model sample uncertainty (both site and measurement errors). 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. To simplify the spatio-temporal model, we assume that a spatial variogram is given. Since a spatial variogram incorporates a spatial variable structure, it allows us to make inference about the temporal unexpectedness.

The simplicity of the method comes from the fact that Gaussian noise as a measurement error, is additive in the OK. Suppose $Y(x)$ represents the measurement of a random spatial field at some spatial point at a specific time. Then the definition of the basic variogram is

$$\gamma_Y = \frac{1}{2}E[(Y(x+h) - Y(x))^2],$$

where $h$ is a distance between two points in the field. Suppose that $Z(x)$ is a noisy version of $Y(x)$. If $N(0,\sigma^2_i)$ and $N(0,\sigma^2_j)$ are additive measurement noise for $x_i, x_j \in \chi$, respectively, the resultant variogram is $\gamma = \gamma_Y + (\sigma^2_i + \sigma^2_j)/2$. Since most robotics work employs formulations lacking such a treatment (cf. [Singh et al., 2010]), the derivation appears in Appendix 6.

Thus, a modified OK framework with different measurement errors can be derived as follows:

$$A_{ij} = \gamma_Y(x_i, x_j) + \frac{1}{2}(\sigma^2_i + \sigma^2_j)$$

$$b_{i0} = \gamma_Y(x_i, x_0) + \sigma^2_i,$$

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

Let $\Delta t_i$ and $\Delta t_j$ be the time elapsed since measuring at $x_i$ and $x_j$, respectively, where $x_i, x_j \in \chi$. We can extend the two measurement errors to a propagated linear function of time, $N(0,\hat{\alpha}\Delta t_i)$ and $N(0,\hat{\alpha}\Delta t_j)$. Here $\hat{\alpha}$ is a scaling parameter for the spatial domain $\chi$.

### 3.3 Online estimation of temporal variability

The dynamics are unlikely to be described adequately by a single value for all time across the whole spatial region, so instead we estimate $\hat{\alpha}$ online using a probabilistic method: a Bayesian filter [Thrun et al., 2005] is maintained for each scale parameter. The value of $\hat{\alpha}$ is modelled via a probability
density function that uses the incoming measurements and the OK estimates over time. This is implemented by keeping a set of values \((\vec{\alpha}, \vec{\omega}) = [(\alpha^1, ..., \alpha^p), (\omega^1, ..., \omega^p)], p \in [1, P]\), where \(\alpha^p\) are temporal scale values, \(\omega^p\) are the (normalized) weights, and \(P\) is the number of the bins. It is natural to visualize \(\omega^p (\sum_{p=[1,P]} \omega^p = 1)\) as vertical lines in a histogram for each \(\alpha^p\). Simply looking at this histogram allows one to interpret the temporal behavior of a region: when comparatively large \(\alpha^p\) has significant probability \(\omega^p\) then the field is fluctuating significantly over time.

To update the temporal variability, the above definitions are substituted into Eq. (5), to produce \(P\) separate modified OK models. The \(p\)th scaled temporal linear function is:

\[
\begin{align*}
A^p_{ij} &= \gamma Y(x_i, x_j) + \frac{1}{2} \alpha^p (\Delta t_i + \Delta t_j) \\
b^p_{i0} &= \gamma Y(x_i, x_0) + \alpha^p \Delta t_i,
\end{align*}
\]

where \(A^p_{ij} \in A^p, b^p_{i0} \in b^p, i, j \in [1, n], p \in [1, P]\).

From Eq. (6), we have \(p\) estimators to update each \(p\)th temporal evolution model. To determine an unobserved spatial location, a single integrated model is constructed via a weighted sum of models. 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 \(\chi\). Then by using Eq. (6), 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 \(KV^*\)):

\[
\begin{align*}
A_{ij} &= \gamma Y(x_i, x_j) + \frac{1}{2} \hat{G}(\Delta t_i + \Delta t_j) \\
b_{i0} &= \gamma Y(x_i, x_0) + \hat{G} \Delta t_i,
\end{align*}
\]

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

This yields the expected value of the evolving field by aggregating each \(\alpha^p\) distribution with weight \(\omega^p\), and \((\vec{\alpha}, \vec{\omega})\) describes the scale of the temporal evolution, modelled as a linear change but with unknown drift.
(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 ($\mu$ 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. **Left**: The increasing size of a circle over time shows the degree of temporal variability of each $\alpha^p$, and how this is captured in the histogram of $\bar{\alpha}$, where the X-axis are scales $\alpha$’s and the Y-axis are weights $\omega$’s. **Right**: Depending on the split rule in Section 4.2, 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 as work is redistributed to balance the total workload.

Figure 1: **Overview of the proposed approach.** Robots represent temporal variability in the field as a form of uncertainty. This is actively estimated by each robot, which is used to (i) further sub-divide if needed, and (ii) determine the work needed per region. The workload is actively balanced across robots by exchanging regions.
4 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. 4.1 and 4.2 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. 4.3.

4.1 Estimation of adaptive field dynamics

This section presents an approach for estimating the temporal evolution shown in Fig. 1(a). As mentioned before, from Eq. (1), (3), and (6), 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 $P(\alpha_t^p, \omega_t^p | \tilde{Z}_{t-1}, Z_t)$, where $\tilde{Z}_{t-1}$ is a past observation set $[Z(x_1),...Z(x_n)]$ and $Z_t$ is the measurement $Z(x_0)$ at time $t$, and $\eta$ is the normalization constant. The algorithm for updating the temporal evolution is shown in Alg. 1.

**Algorithm 1 The Field Dynamics Filter**

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

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

4.2 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 $\chi$ 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 $\chi_i$, where $i \in [1,K]$. Therefore, $\bigcup_{i=1}^{K} \chi_i = \chi$, and $\chi_i \cap \chi_j = \emptyset$, $i \neq j$, and $i,j \in [1,K]$. Each $\chi_i$ 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 4.1. 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 $\hat{G}_{\chi_{i1}}, \hat{G}_{\chi_{i2}}, \hat{G}_{\chi_{i3}}$ where $\bigcup_{j=1}^{4} \chi_{ij} = \chi_i$. Based on this notation and definition, we need to find the maximum difference value of the regions and then subdivide the region so as to best represent the temporal evolution; this is shown in Eq. (8). $M_{cc}$ is the Maximizing Cross-Correlation (differentiation) between sub-regions:

$$M_{cc}^i \equiv \max_{b=[1,4]}(\hat{G}_{\chi_i}, \hat{G}_{\chi_{ib}}), \quad \forall i.$$  

(8)

Since the histogram ($\vec{a}, \vec{\omega}$) represents the rate of the change of the spatial field, we want to subdivide the space so that our knowledge for each region is maximized. The region with maximized knowledge will have a histogram with a small variance that have a peak at some $\alpha$. Thus, Eq. (8) determines that the $\chi_i$ region can be separated into four sub-regions $\chi_{i1}, \chi_{i2}, \chi_{i3}, \chi_{i4}$ and $\chi_{i4}$, or maintained within a current region with regard to a new observation.

Until now, we have assumed that the unobserved location $x_0$ is given, and the robot’s next goal position at time $t$ is $x_0$. We must still describe how to attain $x_0$ in the continuous space. We select location $x_0$ from a set of samples $[x_1, x_2, ..., x_Q]$, where $Q$ is the number of potential locations. To generate the query set, assume that we have $\chi_i$, $i \in [1,K]$. The idea is to distribute the number of queries based on the importance of each region. For each region $i$, we represent its dimension, absolute position, previously sampled points, and $\hat{G}_i$. Since $\hat{G}_i$ is the expected value of the rate of change in $\chi_i$, we define the importance of the region $\chi_i$ to be proportional to the area size $|\chi_i| \times \hat{G}_i$. Randomly selected query locations are produced in subregions according to the importance of the field in that subregion.

The number of queries used as potential waypoints in the space is limited by the computational costs incurred. Suppose the field is well represented
by the sub-divided regions and the number of queries is a parameter. Then, in the limit of this parameter, we can simplify a continuous search space efficiently by allocating different numbers of queries to each sub-divided region based on the importance of the region. In this way, regions with greatest temporal change generate more queries, thereby increasing their sample resolution, compared to those which are more static. This implicitly captures the notion of hotspot sampling. Fig. 4 shows the approach.

Finally, we compute 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:

\[
U(r, x_q)_{q=1, Q} = \sum_{i=1, Q, i \neq q} (KV^*(r, x_i | (\tilde{\alpha}, \tilde{\omega})_{t-1}, \tilde{Z}_{t-1}) - KV^*(r, x_q | (\tilde{\alpha}, \tilde{\omega})_{t-1}, \tilde{Z}_{t-1})).
\]

Eq. (9) includes the navigation cost implicitly when the passage of time is computed by the current robot position, where the robot velocity is given. Suppose that one robot has \( K \) sub-regions including \( \hat{G}_k, k = [1, K] \); Alg. 2 details adaptive path planning for this robot.

Lines 4 to 12 in Alg. 2 separates or merges the fields by comparing \( M_{cc}^k \) and the threshold \( \epsilon \). The number of fields can be increased and decreased when we split or merge the fields. If \( M_{cc}^k \) is larger than \( \epsilon \), it is separated. 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. Thus, Alg. 2 adapts the temporal evolution and updates the model while the robot samples a new observation, and then computes the next goal position in continuous time and the reconstructed regions \( \chi' \).

We compute the summation of the \( KV^* \) of the potential queries instead of the integration of the \( KV^* \) of the continuous space. Moreover, 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. If the robot may sample more points during
Algorithm 2 The Adaptive Path Planner

1: INPUT: \((\tilde{a}_t, \tilde{\omega}_t)_k, \forall k, r, a\) split condition \(\epsilon\) and a total number of queries \(Q\).
2: OUTPUT: \(x_0\) and \(\chi'\).
3: \(\chi' = G' = \text{NULL}\)
4: for \(k = 1\) to \(K\) do
5:   if \(M_{cc}^k > \epsilon\) then
6:     \(\chi' = \chi' + [\chi_{k1}, \chi_{k2}, \chi_{k3}, \chi_{k4}]\)
7:     \(\hat{G}' = \hat{G}' + [\hat{G}_{\chi_{k1}}, \hat{G}_{\chi_{k2}}, \hat{G}_{\chi_{k3}}, \hat{G}_{\chi_{k4}}]\)
8:   else
9:     \(\chi' = \chi' + [\chi_k]\)
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:    \(N_k = Q \times |\chi'_k| \times \hat{G}'_k \div \text{factor}\)
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 next goal position \(x_0\), following \(\max(U_q)\), and new regions \(\chi'\) with \(\hat{G}'\)
its movement to the goal position with no additional measurement cost (since the $KV^*$ already includes navigation cost) then the quality of the prediction will be improved further.

4.3 Distributed multi-robot sampling

Regions also have an associated graphical representation. A region $\chi$ 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 $\chi_k$, 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 $\chi$. 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 $\chi$ and $N$ number of edges, each $w^v_i$ computed by $|\chi_a| \times \hat{G}_a$ is the importance of the region, $a \in [1, L]$. $w^e_{ab}$ is the distance between $v_{a1}$ and $v_{a2}$, $b \in [1, N]$ and $a_1, a_2 \in [1, L]$. Then this problem can be formulated as the following optimization problem:

$$\arg \min_{(G_i, G_j)} \left[ \max_{i \in [1, m]} \left( \sum_{a,b} (w^v_{a1} + w^e_{ab}) \right) - \min_{j \in [1, m]} \left( \sum_{c,d} (w^v_{c2} + w^e_{cd}) \right) \right].$$  \hspace{1cm} (10)$$

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 [Garey et al., 1976], we adopt the heuristic of [Kernighan and Lin, 1970] because of its good performance in adjusting the unbalanced partitions as well as its practical running speed $O(n^3)$. A modification is necessary in our setting because robot may need to pass by neighboring regions in order to reach its final goal position. Thus, 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 4.2, 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 $\sum w^i_v + \sum w^i_e$, 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: $G'_i = G_i - \{v_a\}$
19: $G'_j = G_j + \{v_a\}$
5 Simulation and experiments

5.1 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 [National Oceanic and Atmospheric Administration 2012], 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 [NASA’s Moderate Resolution Imaging Spectroradiometer 2012]) 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) [National Oceanic and Atmospheric Administration 2012] is collected by fixed buoys which are spread across large areas (e.g., 3 buoys per 100 \( km^2 \)).

To successfully run the experiments, we need to use dense data. We interpolate chlorophyll data for Lake Erie using the daily MODIS [NASA’s Moderate Resolution Imaging Spectroradiometer 2012] data set (June to August 2012). But since we are focused on maximizing the quality of an estimate of the temporal dynamics of a spatially inhomogeneous process, we opted to supplement the base field with a contaminant plume. We employed a Gaussian plume model to simulate data from a dense plume where, using the two axes of diffusivity and velocity, one can produce dynamics similar to what one might see in the ocean or a lake [Holzbecher 2007].

A Gaussian Plume model for transient transport, including diffusion and dispersion, is given by Eq. (11), \( M \) is mass of the plume, where \( D_x \) and \( D_y \) are the diffusivity 313° 5 m/s, \( v_x \) and \( v_y \) are the wind speed 269° 1 m/s, and \( t \) is the time scale. These parameters are selected by considering the current fixed buoys [National Oceanic and Atmospheric Administration 2012].

\[
\frac{M}{4\pi t \sqrt{D_x D_y}} \exp\left(-\frac{1}{4t} \left(\frac{(x - v_x t)^2}{D_x} + \frac{(y - v_y t)^2}{D_y}\right)\right) + N(0, \sigma^2).
\] (11)

For the experiments we generated an interpolation of the data for 1000 time 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:
Figure 2: This shows the plume-like ground truth data. (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 (c).
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 $5 - 7 \text{ m/s}$. The greedy planner, which is selecting the next goal position in order to minimize the total uncertainty without considering the temporal changes [Kim et al., 2012], 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.

5.2 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 69th step. Fig. 3(b) represents 13 different field dynamics after the 217th 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.

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 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. 6 demonstrates the distributed workloads for two robots. After a
Figure 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 $\alpha^p$ and the Y-axis is $\omega^p$. 
Figure 4: The initial temporal evolution weight set is an uniform distribution, and then updated when the robot samples new data. If $E_X$ is enough to split or merge, each region has different $\omega$ distribution depends on the given data. X-axis is $\alpha^p$, and Y-axis is $\omega^p$. 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 histogram of the region.
Figure 5: Two planners that are cross-validations with ground truth. One is between predictions based on our approach and ground truth over time. Another is between predictions based on the traditional approach and the ground truth over time. Data are from five separate trials of each case. This shows mean and variance over time.
(a) Distributed workload for two robots at 454th step

(b) Overall workload distribution over time

Figure 6: The workload is distributed to two robots while each robot estimates and updates their field dynamics. 6(a) shows that the graph represents the workload including field dynamics, the size of each region, and the distance between nearby regions. 6(b) shows several sharp increases when the transition of separation is happened in the region (e.g., 75th, 140th, 200th, 350th, and 450th steps).
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\textsuperscript{th} step, the workload increases for a while with the field redistribution by Alg. 3, but when another transition happens at the 450\textsuperscript{th} step, the workload gradually decreases. In other words, Fig. 6 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.

6 Conclusion

Environmental robotic sampling is a challenging task since the phenomena of interest may change with time in ways that are dependant on space. 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 [Cressie and Wikle, 2011], 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 [Garg et al., 2012] can address this, but it remains unclear how such an implementation might be distributed in communication constrained settings.

This paper considers the case where temporal behavior is a function of space. However, it is conceivable that spatial behavior is either a function of space or a function of time. We expect that similar methods based on either space or time will address these cases which are beyond the scope of this paper.

Many researchers have tried to detect and track the hotspots in the field such as plumes, pollution, and volcanic activity. We believe that our approach can help detect and track hotspots and can allow for sampling a field over time when the field has non-homogeneous temporal dynamics. The approach proposed here is also applicable to underwater and surface vehicles as well as aerial vehicles and ground vehicles. Ultimately, the modeling of field dynamics enhances the quality of environmental sampling and monitoring for multi-robot systems.

Acknowledgement

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

Appendix: Field dynamics in variogram

The following theorem relates two different Gaussian noise model in Ordinary Kriging in Eq. 7.

**Theorem 1** Let \( \gamma^Y \) be a basic variogram treated as a parameter of the random process \( Y(x) \). Suppose \( Z(x) \) is a noisy version of \( Y(x) \), assume two points \( x_i \) and \( x_j \in \chi \) have two different Gaussian noise models, \( \epsilon_i \sim N(0, \sigma_i^2) \) and \( \epsilon_j \sim N(0, \sigma_j^2) \). The variogram with two independent Gaussian noise model is \( \gamma(h) = \gamma^Y(h) + \frac{1}{2}(\sigma_i^2 + \sigma_j^2) \).
**Proof:** We have $\epsilon_i - \epsilon_j = \tilde{Y} \sim N(0, \sigma_i^2 + \sigma_j^2)$ and $|x_i - x_j| = h$ where $x_i > x_j$. Consider $\gamma(h) = \frac{1}{2} E[(Z(x_i) - Z(x_j))^2]$.

\[
\begin{align*}
\gamma(h) &= \frac{1}{2} E[(Y(x_i) - Y(x_j) + \epsilon_i - \epsilon_j)^2] \\
 &= \frac{1}{2} E[(Y(x_i) - Y(x_j) + \tilde{Y})^2] \\
 &= \frac{1}{2} E[(Y(x_i) - Y(x_j))^2 + 2(Y(x_i) - Y(x_j))(\tilde{Y}) + \tilde{Y}^2] \\
 &= \frac{1}{2} E[(Y(x_i) - Y(x_j))^2] + E[(Y(x_i) - Y(x_j))(\tilde{Y})] + \frac{1}{2} E[\tilde{Y}^2] \\
 &= \gamma^Y(h) + E[(Y(x_i) - Y(x_j)) \cdot \tilde{Y}] + \frac{1}{2}(\sigma_i^2 + \sigma_j^2) \\
 &= \gamma^Y(h) + \frac{1}{2}(\sigma_i^2 + \sigma_j^2)
\end{align*}
\]

Thus, the variogram with two different Gaussian noise models, $\frac{1}{2} E[(Z(x_i) - Z(x_j))^2]$ is simplified by $\gamma(h) = \gamma^Y(h) + \frac{1}{2}(\sigma_i^2 + \sigma_j^2)$. 