Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

A cooperative deployment strategy for optimal sampling in spatiotemporal estimation

Rishi Graham   Jorge Cortés

Abstract— **This paper considers a network composed of robotic agents and static nodes performing spatial estimation of a dynamic physical process. The physical process is modeled as a spatiotemporal random field with finite spatial correlation range. We propose a distributed coordination algorithm to optimize data acquisition across time. The robotic agents take measurements of the process and relay them to the static nodes. The static nodes collectively compute directions of maximum descent of the estimation uncertainty, and relay them back to the robotic agents. The technical approach combines tools from geostatistics, parallel computing, and systems and control. We illustrate the soundness of the algorithm in simulation.**

I. Introduction

Problem statement: This paper considers a network of static nodes and robotic sensors taking sequential measurements of a dynamic physical process. We model the underlying process as a spatiotemporal random field. Our objective is to determine trajectories for the robots which optimize data acquisition in order to best estimate the field. This problem has applications in environmental monitoring, oceanographic surveying, and atmospheric sampling.

Literature review: Kriging [1], [2] is a standard geostatistical technique for estimating spatiotemporal random fields. Given a set of point measurements, kriging produces a predictor of the field throughout the environment, along with a measure of the associated uncertainty. Under certain conditions on the covariance, data taken far from the prediction site have very little impact on the kriging predictor [3]. When the spatiotemporal random field does not have a finite covariance range, an approximation may be generated via covariance tapering [4]. The optimal design literature [5], [6] deals with the problem of determining sets of locations where data should be taken in order to optimize the resulting estimation. The work [7] examines the effect that adding and deleting measurement locations has on the kriging variance, and how this relates to optimal design.

The field of cooperative control for mobile sensor networks has received much recent attention. [8] introduces performance metrics for oceanographic surveys by autonomous underwater vehicles. [9] considers a network of robotic sensors with centralized control estimating a static field from measurements with both sensing and localization error. [10] considers choosing the optimal sampling trajectories from a parameterized set of paths. In [11], [12] the focus is on estimating deterministic fields when the measurements taken by individual robots are uncorrelated. [13] discusses the tracking of level curves in a noisy scalar field.

Statement of contributions: We consider a robotic network comprised of static nodes and mobile sensor agents. This combination allows us to distribute the burden associated with sensing, communication, and computing. The environment is partitioned into regions, and each static node is responsible of maintaining an approximation of the spatial field on its region. The nodes are deployed so that their communication topology is connected, and any robotic agent can communicate to at least one node at any given time. The robots are responsible for taking measurements of the field and relaying them back to the nearest nodes.

The main contribution of this paper is the design of a distributed coordination algorithm to optimally sample dynamic physical processes modeled as spatiotemporal random fields. As a criterion for optimality, we consider the spatiotemporal average of the kriging variance. This function has the natural interpretation of an aggregate objective function that measures the uncertainty about the knowledge of the random field. Under the assumption of a finite correlation range in space, we provide an upper bound on the kriging variance, which in turn induces an upper bound on our objective function amenable to distributed optimization. The static nodes compute the gradient of the approximate average kriging variance and relay simple control vectors back to the robots. This guarantees that the next measurements are taken at positions which decrease the approximate overall uncertainty of the estimation. We do not pay attention to how the estimation is actually implemented, but focus on minimizing the uncertainty of the estimate so that data acquisition is optimized. Proofs are omitted and will appear elsewhere.

Organization: Section II introduces basic notation and tools from constrained optimization and kriging estimation. Section III introduces the robotic network model and details the overall network objective. The following two sections present important ingredients in the ulterior algorithm design. Section IV specifies the regions of allowed motion for the robotic agents at each step, while Section V describes an upper bound of the spatiotemporal average of the kriging variance. Section VI presents the distributed coordination algorithm that the robotic network executes to optimize data acquisition, along with some illustrative simulations. Section VII contains our conclusions and ideas for future work.

II. Preliminary notions

Let $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$ denote the set of reals, positive reals and nonnegative reals, respectively. For $p \in \mathbb{R}^d$ and $r \in \mathbb{R}_{>0}$, let $\overline{B}(p, r)$ be the *closed ball* of radius r centered at p. Given two vectors $u = (u_1, \dots, u_a)^T$, $a \in \mathbb{Z}_{>0}$, and $v = (v_1, \dots, v_b)^T$, $b \in \mathbb{Z}_{>0}$, we denote by (u, v) its concatenation $(u, v) = (u_1, \dots, u_a, v_1, \dots, v_b)^T$. We denote

R. Graham is with the Department of Applied Mathematics and Statistics, University of California, Santa Cruz, rishig@ams.ucsc.edu

J. Cortés is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, cortes@ucsd.edu

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

by ∂S the boundary of a set S. The *ϵ-contraction* of a set S, with $\epsilon > 0$, is the set $S_\epsilon = \{q \in S \mid \mathrm{d}(q, \partial S) \geq \epsilon\}$. A *convex polytope* is the convex hull of a finite point set. For a bounded set $S \subset \mathbb{R}^d$, we let $\mathrm{CR}(S)$ denote the *circumradius* of S, that is, the radius of the smallest-radius d-sphere enclosing S. We denote by $\mathbb{F}(S)$ the collection of finite subsets of S.

We consider a compact and connected set $\mathcal{D} \subset \mathbb{R}^d$, $d \in \mathbb{N}$. Since we deal with a process which varies over time, let $\mathcal{D}_e = \mathcal{D} \times \mathbb{R}$ denote the space of points over $\mathcal{D}$ and time.

The *Voronoi partition* $\mathcal{V}(\boldsymbol{s}) = (V_1(\boldsymbol{s}), \dots, V_n(\boldsymbol{s}))$ of $\mathcal{D}$ generated by the points $\boldsymbol{s} = (s_1, \dots, s_n)$ is defined by

$$V_i(\boldsymbol{s}) = \{q \in \mathcal{D} \mid \|q - s_i\| \leq \|q - s_j\|, \, \forall j \neq i\} .$$

Each $V_i(\boldsymbol{s})$ is called a *Voronoi cell*. Two points s_i and s_j are *Voronoi neighbors* if their Voronoi cells share a boundary.

A. Projected gradient descent

Next, we describe the constrained optimization technique known as projected gradient descent [14] to iteratively find the minima of an objective function $F : \mathbb{R}^m \to \mathbb{R}_{\geq 0}$. Let Ω denote a nonempty, closed, and convex subset of $\mathbb{R}^m$, $m \in \mathbb{N}$. Assume that ∇F is globally Lipschitz on Ω. Let $\mathrm{proj}_\Omega : \mathbb{R}^m \to \Omega$ denote the orthogonal projection onto the set Ω,

$$\mathrm{proj}_\Omega(x) = \operatorname*{argmin}_{y \in \Omega} \|x - y\|.$$

Consider a sequence $\{x_k\} \in \Omega$, $k \in \mathbb{N}$, which satisfies

$$x_{k+1} = \mathrm{proj}_\Omega \left(x_k - a_k \nabla F(x_k) \right), \, x_1 \in \Omega, \qquad (1)$$

where the step size, a_k, is chosen according to the LINE SEARCH ALGORITHM described in Table I, evaluated at $x = x_k$.

Name:	LINE SEARCH ALGORITHM
Goal:	Determine step size for algorithm (1)
Input:	$x \in \Omega$
Assumes:	$\tau, \theta \in (0, 1)$, max step $\alpha_{\max} \in \mathbb{R}_{>0}$
Output:	$\alpha \in \mathbb{R}_{\geq 0}$

```
1: $\alpha = \alpha_{\max}$
2: repeat
3:     $x_{\text{new}} = \mathrm{proj}_\Omega \left( x - \alpha \nabla F(x) \right)$
4:     $\nu = \frac{\theta}{\alpha} \|x - x_{\text{new}}\|^2 + F(x_{\text{new}}) - F(x)$
5:     if $\nu > 0$ then
6:         $\alpha = \alpha\tau$
7: until $\nu \leq 0$
```

TABLE I

LINE SEARCH ALGORITHM.

With $\theta > 0$, the LINE SEARCH ALGORITHM must terminate in finite time. Increasing θ decreases the number of iterations. The Armijo condition (step 7) ensures that the decrease in F is commensurate with the magnitude of its gradient. A sequence $\{x_k\}_{k=1}^\infty$ satisfying these requirements converges in the limit as $k \to \infty$ to stationary points of F [14].

B. Estimation via Kriging interpolation

This section reviews the geostatistical kriging procedure for estimating spatial processes, see e.g., [1], [2]. We assume that the random process Z is of the form

$$Z(h) = \mu(h) + \delta(h), \quad h \in \mathcal{D}_e, \qquad (2)$$

with mean μ, and δ a zero-mean second-order stationary random process with a known covariance function $C : \mathcal{D}_e \times \mathcal{D}_e \to \mathbb{R}_{\geq 0}$ that has a *finite spatial range* $r \in \mathbb{R}_{>0}$,

$$C\left((s_1, t_1), (s_2, t_2)\right) = 0, \text{ if } \|s_2 - s_1\| > r.$$

We assume the data, $\boldsymbol{y} = y(\boldsymbol{h}) = (Y(h_1), \dots, Y(h_l))^T$, are corrupted with errors,

$$Y(h_i) = Z(h_i) + \epsilon_i, \qquad \epsilon_i \overset{\text{iid}}{\sim} N\left(0, \sigma_\epsilon^2\right), \, \sigma_\epsilon \in \mathbb{R}. \qquad (3)$$

The constant variance in measurement error models identical sensors. The covariance between $Y(h_i)$ and $Y(h_j)$ is written

$$\mathrm{Cov}[Y(h_i), Y(h_j)] = \begin{cases} C(h_i, h_j) + \sigma_\epsilon^2, & \text{if } i = j, \\ C(h_i, h_j), & \text{otherwise.} \end{cases}$$

Let $c : \mathcal{D}_e \times \mathcal{D}_e^l \to \mathbb{R}^l$ denote the vector of covariances between $Z(h)$, $h \in \mathcal{D}_e$ and $\boldsymbol{y}$, and let $\Sigma : \mathcal{D}_e^l \to \mathbb{R}^{l \times l}$ denote the covariance matrix of $\boldsymbol{y}$.

Kriging aims to minimize the error variance,

$$\sigma^2(Z(h); \boldsymbol{h}) = \mathrm{Var}\left[Z(h) - \mathrm{pred}(Z(h); \boldsymbol{h})\right], \qquad (4)$$

of the predictor, $\mathrm{pred}(Z(h); \boldsymbol{h})$, of Z at a location $h \in \mathcal{D}_e$ from data measured at locations $\boldsymbol{h}$. As we are primarily concerned with estimation uncertainty, we omit explicit representation of the predictors themselves (see, e.g. [2] for details). Assuming μ is known, the *simple kriging predictor*, $\hat{z}_{\mathrm{SK}}(h; \boldsymbol{h})$, minimizes (4) among unbiased predictors of the form $\mathrm{pred}(Z(h); \boldsymbol{h}) = \sum_{i=1}^l \alpha_i Y(h_i) + k$. The error variance of $\hat{z}_{\mathrm{SK}}(h; \boldsymbol{h})$ is,

$$\sigma_{\mathrm{SK}}^2(Z(h); \boldsymbol{h}) = \sigma_Z^2(h) - \boldsymbol{c}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{c}. \qquad (5)$$

Here $\sigma_Z^2(h) = C(h, h)$ denotes the variance of $Z(h)$, while $\boldsymbol{c}\boldsymbol{\Sigma}^{-1}\boldsymbol{c}$ represents the variance of $\hat{z}_{\mathrm{SK}}(h; \boldsymbol{h})$.

Relaxing the assumption that μ is known, consider a linear expansion $\mu(h) = f(h)^T \beta$, where $f = (f_1, \dots, f_p)^T : \mathcal{D}_e \to \mathbb{R}^p$ is known and $\beta = (\beta_1, \dots, \beta_p)^T \in \mathbb{R}^p$ is unknown. The *universal kriging predictor*, $\hat{z}_{\mathrm{UK}}(h; \boldsymbol{h})$, minimizes (4) among all unbiased predictors of the form $\mathrm{pred}(Z(h); \boldsymbol{h}) = \sum_{i=1}^l \alpha_i Y(h_i)$, with error variance,

$$\sigma_{\mathrm{UK}}^2(Z(h); \boldsymbol{h}) = \sigma_Z^2(h) - \boldsymbol{c}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{c} + \Big(f - \boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{c}\Big)^T \Big(\boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{F}\Big)^{-1} \Big(f - \boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{c}\Big), \qquad (6)$$

where $\boldsymbol{F}$ denotes the matrix whose ith row is $f(h_i)^T$. Unless stated otherwise, the results make no distinction between simple and universal kriging. To simplify notation, we drop the subscript and use $\hat{z}$ to denote both estimators, with associated variance σ^2. Since σ^2 is invariant under permutations, we evaluate it at a set, instead of at a tuple. We also denote by $\hat{y}$ the estimate of the noisy measurement Y, identical to $\hat{z}$ except in the extra constant term σ_ϵ^2.

III. PROBLEM STATEMENT

A. Robotic sensor network model

Consider a group $\{S_1, \dots, S_m\}$ of $m \in \mathbb{N}$ static nodes with limited communication radius, $R \in \mathbb{R}_{>0}$, deployed in a convex polytope $\mathcal{D} \subset \mathbb{R}^d$ at positions $Q = (q_1, \dots, q_m) \in \mathcal{D}^m$. Assume that they are positioned so that each one can communicate with its Voronoi neighbors.

In addition to the static nodes, consider a group $\{R_1, \dots, R_n\}$ of n robotic sensor agents. The position of robot $i \in \{1, \dots, n\}$ at time $t \in \mathbb{R}$ is denoted by $p_i(t) \in \mathcal{D}$. We assume that robots take measurements of the spatial field at discrete instants of time in $\mathbb{Z}_{\geq 0}$. Between measurement instants, each robot moves according to the discrete dynamics

$$p_i(k+1) = p_i(k) + u_i(k),$$

where $\|u_i\| \leq u_{\max}$ for some $u_{\max} \in \mathbb{R}_{>0}$. The communication radius of the robotic agents is also R. Each node needs to communicate with any robot which may be within covariance range of the points in its Voronoi region at the following timestep. To that end, we assume that

$$R \geq \max_{i \in \{1,\dots,m\}} \{\mathrm{CR}(V_i(Q))\} + r + u_{\max}. \qquad (7)$$

Assume that each robot can sense the positions of any other robots within a distance of $2u_{\max}$. At discrete timesteps, each robot communicates the measurement and location to static nodes within communication range, along with the locations of any other sensed robots. The nodes are responsible for calculating control-specific information and relaying it back to those robots within communication range. Our implementation does not require direct communication between robots. We refer to this network model as $\mathcal{N}$.

B. The average kriging variance as objective function

Our objective is to design a strategy to optimize the estimation of Z. We encode this objective into the optimization of an aggregate function that we describe next. Assume the experiment has run for $k_{\max} \in \mathbb{Z}_{\geq 0}$ timesteps and a sequence of measurements taken at time intervals $\{1, \dots, k_{\max}\}$, at space-time locations $\boldsymbol{h} \in (\mathcal{D}_e^n)^{k_{\max}}$ are available. Consider a kriging estimation $\hat{z}(h; \boldsymbol{h})$ made on $\mathcal{D}$ over $T = [1, k_{\max}]$. The average error variance of $\hat{z}(h; \boldsymbol{h})$ is

$$\mathcal{A} = \int_T \int_{\mathcal{D}} \sigma^2(Z(s,t); \boldsymbol{h})\, ds\, dt.$$

One would like to choose the measurement locations that minimize $\mathcal{A}$. Since measurements are taken sequentially, and each set is restricted to a region nearby the previous measurements, one cannot simply optimize over $(\mathcal{D}_e^n)^{k_{\max}}$. Additionally, $k_{\max}$ may not be known.

Consider, instead, a greedy approach in which we use past measurements to choose the positions for the next ones. Let $\boldsymbol{h}^{(\leq k)} \in (\mathcal{D}_e^n)^k$ be the vector of measurement location and time pairs for timesteps up to k. Let $(P, k+1)$ denote the space-time locations at spatial positions $P = (p_1, \dots, p_n) \in \mathcal{D}^n$ and time $k+1$. Let $\mathcal{A}^{(k)} : \mathcal{D}^n \to \mathbb{R}$ be defined as,

$$\mathcal{A}^{(k)}(P) = \int_T \int_{\mathcal{D}} \sigma^2 \left(Z(s,t); (\boldsymbol{h}^{(\leq k)}, (P, k+1)) \right) ds\, dt.$$

The objective is to choose the set of measurement locations P to maximally decrease $\mathcal{A}^{(k)}$. Unfortunately, the gradient of $\mathcal{A}^{(k)}$ cannot be computed in a distributed way by the static nodes because of the matrix inversions which depend on all measurement locations. Instead, our approach is to construct an upper bound to $\mathcal{A}^{(k)}$ whose gradient is distributed and design an algorithm to optimize it.

IV. Voronoi contraction for collision avoidance

We begin by specifying the region of allowed movement for the robotic agents. In addition to the maximum velocity and the requirement of staying within $\mathcal{D}$, we impose a minimum distance requirement between robots. Beyond the benefit of collision avoidance, this restriction ensures that even under the assumption of zero sensor error, the kriging error function is well-defined over the space of possible configurations.

Let $\omega \in \mathbb{R}_{\geq 0}$ be a desired buffer width, assumed to be small compared to the size of $\mathcal{D}$. To ensure that the distance between two robots is never smaller than ω, we introduce a contraction of the Voronoi diagram. Consider the locations $P = (p_1, \dots, p_n)$ of the n robotic agents at the kth timestep. Let $\Omega_i^{(k)} \subset \mathcal{D}$ such that $\Omega_i^{(k)} = (V_i(P))_{\omega/2} \cap \overline{B}(p_i, u_{\max})$, where $(V_i(P))_{\omega/2}$ denotes the $\frac{\omega}{2}$-contraction of $V_i(P)$. For each $j \neq i \in \{1, \dots, n\}$, we have $\mathrm{d}(\Omega_i^{(k)}, \Omega_j^{(k)}) \geq \omega$. Between timesteps k and $k+1$, we restrict R_i to the region $\Omega_i^{(k)}$. Figure 1 shows an example in $\mathbb{R}^2$ of this set.

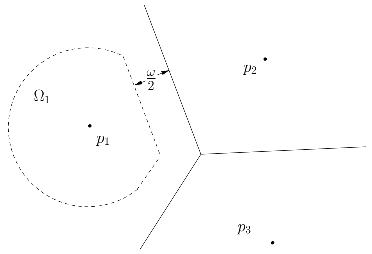

Fig. 1. Regions $\{\Omega_i^{(k)}\}_{i=1}^n$ (dashed) versus Voronoi partition (solid).

Let $\Omega^{(k)} = \prod_{i=1}^n \Omega_i^{(k)} \subset (\mathbb{R}^d)^n$ denote the region of allowed movement of all robotic agents at timestep $k \in \mathbb{Z}_{\geq 0}$. Note that $\Omega^{(k)}$ is closed, bounded, and convex.

V. Approximate average kriging variance

Here we compute an upper bound on the average kriging variance. We begin by providing a useful result that isolates the effect of a subset of measurements on the kriging variance. We need the following notation for predicting Y at vectors of locations: given $\boldsymbol{h}_1 \in \mathcal{D}_e^l$ and $\boldsymbol{h}_2 = (h_{21}, \dots, h_{2m})^T \in \mathcal{D}_e^m$, with $i_{\mathbb{F}}(\boldsymbol{h}_1) \cap i_{\mathbb{F}}(\boldsymbol{h}_2) = \emptyset$, let

$$\hat{y}(\boldsymbol{h}_2; \boldsymbol{h}_1) = \hat{y}_{21} = [\hat{y}(h_{21}; \boldsymbol{h}_1), \dots, \hat{y}(h_{2m}; \boldsymbol{h}_1)]^T,$$
$$\bar{y}(\boldsymbol{h}_2; \boldsymbol{h}_1) = \bar{y}_{21} = y(\boldsymbol{h}_2) - \hat{y}(\boldsymbol{h}_2; \boldsymbol{h}_1).$$

We are ready to present an upper bound on the error variance.

Lemma V.1 (Upper bound on kriging variance) *Let $\boldsymbol{h} = (\boldsymbol{h}_1, \boldsymbol{h}_2)$ denote a full set of distinct measurement locations, with $\boldsymbol{h}_1 = (h_1, \dots, h_l) \in \mathcal{D}_e^l$ and $\boldsymbol{h}_2 = (h_{l+1}, \dots, h_n) \in \mathcal{D}_e^m$, with $l + m = n$. Then,*

$$\sigma^2(Z(h); \boldsymbol{h}) = \sigma^2(Z(h); \boldsymbol{h}_1) - \mathrm{Cov}[\bar{z}(h; \boldsymbol{h}_1), \bar{y}_{21}] \cdot$$
$$\cdot \mathrm{Var}[\bar{y}_{21}]^{-1} \mathrm{Cov}[\bar{y}_{21}, \bar{z}(h; \boldsymbol{h}_1)] \leq \sigma^2(Z(h); \boldsymbol{h}_1),$$

with equality if $\mathrm{Cov}[Z(h), y(\boldsymbol{h}_2)] = 0 = \mathrm{Cov}[y(\boldsymbol{h}_1), y(\boldsymbol{h}_2)]$.

Note that $\sigma^2(Z(h); \boldsymbol{h}_1)$ corresponds to the error variance of a predictor computed with the information at locations $\boldsymbol{h}_1$.

Proposition V.2 (Spatial approximation for distributed implementation) *Define* $\mathit{CS}^{(\leq k+1)} : \mathbb{Z}_{\geq 0} \times \mathcal{D}^n \to \mathbb{F}(\mathcal{D}_e)$ *by*

$$\mathit{CS}^{(\leq k+1)}(j, P) = \\ \Big\{(s,t) \in i_{\mathbb{F}}\left(\boldsymbol{h}^{(\leq k)}, (P, k+1)\right) \mid \mathrm{d}(s, V_j(Q)) \leq r\Big\},$$

i.e., the subset of measurement locations up to time $k+1$ *which are correlated in space to the Voronoi cell of the static node* j*. Let* $\tilde{\mathcal{A}}_j^{(k)} : \mathcal{D}^n \to \mathbb{R}$ *be defined by*

$$\tilde{\mathcal{A}}_j^{(k)}(P) = \int_T \int_{V_j(Q)} \sigma^2\left(Z(s,t); \mathit{CS}^{(\leq k+1)}(j,P)\right) ds\, dt.$$

Then $\mathcal{A}^{(k)} \leq \sum_{j=1}^m \tilde{\mathcal{A}}_j^{(k)}$*. In addition, equality holds if, for all* $j \in \{1, \ldots, m\}$*, the points in* $\mathit{CS}^{(\leq k+1)}(j, P)$ *are not correlated to other measurement locations outside it.*

Remark V.3 ($\tilde{\mathcal{A}}_j^{(k)}$ may be calculated with local information only) The location p_i contributes only to $\tilde{\mathcal{A}}^{(k)}(P)$ in those Voronoi regions $V_j(Q)$ for which $\mathrm{d}(p_i, V_j(Q)) \leq r$. Thus the requirement (7) ensures that S_j can calculate $\tilde{\mathcal{A}}_j^{(k)}$. As with σ^2, we evaluate $\tilde{\mathcal{A}}_j^{(k)}$ at a set, rather than a tuple.•

Remark V.4 (Universal kriging with too few measurements) It should be noted here that in the universal kriging case the function $\tilde{\mathcal{A}}_j^{(k)}$ is only well-defined if the number of measurement locations available to each node is greater than or equal to the number of basis functions. In this paper we assume that this holds at all times. •

Our next step is to characterize the smoothness properties of $\tilde{\mathcal{A}}^{(k)}$. Let us introduce some notation. For $i \in \{1, \ldots, n\}$,

$$\mathrm{CS}_{-i}^{(\leq k+1)}(j, P) = \mathrm{CS}^{(\leq k+1)}(j, P) \setminus \{(p_i, k+1)\}.$$

Define the maps $C_i : \{1, \ldots, m\} \times \mathcal{D}^n \times \mathcal{D} \times T \to \mathbb{R}$ and $V_i : \{1, \ldots, m\} \times \mathcal{D} \to \mathbb{R}$ by

$$\begin{aligned} C_i(j, P, s, t) = \mathrm{Cov}[&\bar{y}((p_i, k+1), \mathrm{CS}_{-i}^{(\leq k+1)}(j, P)), \\ &\bar{z}((s,t), \mathrm{CS}_{-i}^{(\leq k+1)}(j, P))], \\ V_i(j, P) = \mathrm{Var}[&\bar{y}((p_i, k+1), \mathrm{CS}_{-i}^{(\leq k+1)}(j, P))], \end{aligned}$$

where $\bar{y}(h; \boldsymbol{h}) = Y(h) - \hat{y}(h; \boldsymbol{h})$ is a shorthand notation. For $s \in \mathcal{D}$ and $t \in T$, let $\nabla_i C_i(j, P, s, t)$ and $\nabla_i V_i(j, P)$ denote the partial derivative of C_i and V_i with respect to p_i. Next, we provide an expression for the gradient of $\tilde{\mathcal{A}}^{(k)}$.

Proposition V.5 *Assume that the covariance of* Z *is* C^1 *with respect to the spatial position of its arguments. For universal kriging, further assume that* $f_1, \ldots, f_p$ *are* C^1 *with respect to the spatial position of their arguments. Then* $\tilde{\mathcal{A}}^{(k)}$ *is* C^1 *on* $\Omega^{(k)}$ *and the ith component of its gradient is*

$$\nabla_i \tilde{\mathcal{A}}^{(k)}(P) = \sum_{j=1}^m \nabla_i \tilde{\mathcal{A}}_j^{(k)}(P),$$

$$\nabla_i \tilde{\mathcal{A}}_j^{(k)}(P) = \frac{\int_T \int_{V_j(Q)} C_i(j,P,s,t)^2\, ds\, dt\, \nabla_i V_i(j,P)}{V_i(j,P)^2} \\ - \frac{2 \int_T \int_{V_j(Q)} C_i(j,P,s,t) \nabla_i C_i(j,P,s,t)\, ds\, dt}{V_i(j,P)}.$$

For each $i \in \{1, \ldots, n\}$, $\nabla_i \tilde{\mathcal{A}}_j(P)$ may be computed by node j, and thus $\nabla_i \tilde{\mathcal{A}}^{(k)}(P)$ may be computed in a distributed way on the network of nodes. The next result characterizes the global Lipschitzness of $\nabla \tilde{\mathcal{A}}^{(k)}$.

Proposition V.6 *Under the assumptions of Proposition V.5, make the following additional assumptions,*

- $\nabla_i \mathrm{Cov}[Z(p_i, k+1), Z(s_2, t_2)]$ *is globally Lipschitz on* $\Omega_i^{(k)}$ *for each* $i \in \{1, \ldots, n\}$*;*
- *in the universal kriging case, further assume that the partial derivatives* $\frac{\partial}{\partial s} f_j$ *are globally Lipschitz on* $\Omega_i^{(k)}$*.*

Then the gradient $\nabla \tilde{\mathcal{A}}^{(k)}$ *is globally Lipschitz on* $\Omega^{(k)}$*.*

VI. Optimizing information retrieval via gradient descent

In this section, we design a coordination algorithm to follow the gradient of $\tilde{\mathcal{A}}^{(k)}$. We consider a system in which each static node is responsible for calculating control vectors for the robotic agents within the region of influence. We use the formulation of the approximate average error presented in Proposition V.2, and follow a projected gradient descent building on Section II. The current timestep, k, is held fixed through the section and, to reduce notation, we leave off the superindex which indicates timestep where unnecessary. $P = (p_1, \ldots, p_n)$ denotes the current positions of the robots.

A. Distributed optimization of the approximate variance

Ideally, at the kth timestep, we would like the robots to move to the minimum of $P \mapsto \tilde{\mathcal{A}}(P)$. Finding such a minimum over the whole region is a difficult problem. Instead, we use a distributed version of the projected gradient descent algorithm, which is guaranteed to converge to a stationary point. For convenience, we define the following notation. Let $P'_j : \mathbb{R} \times \mathcal{D}^n \to \mathbb{F}(\mathcal{D})$ map a step size and a configuration to the set of next locations calculated by S_j,

$$P'_j(\alpha, P) = \Big\{\mathrm{proj}_{\Omega_i}\left(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P)\right), \\ \text{foreach } i \text{ s.t. } \mathrm{d}\left(p_i, V_j(Q)\right) \leq r + u_{\mathrm{max}} + \omega\Big\}.$$

Let $\mathrm{d}_j : \mathbb{R} \times \mathcal{D}^n \to \mathbb{R}_{\geq 0}$ denote the total distance traveled by robots entering $V_j(Q)$, i.e.,

$$\mathrm{d}_j\left(\alpha, P\right) = \sum_{\substack{i \in \{1,\ldots,n\} \text{ such that} \\ \mathrm{proj}_{\Omega_i}\left(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P)\right) \in V_j(Q)}} \| \mathrm{proj}_{\Omega_i}\left(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P)\right) - p_i \|^2.$$

Globally, let $P' : \mathbb{R} \times \mathcal{D}^n \to \mathcal{D}^n$, $P'(\alpha, P) = \mathrm{proj}_{\Omega}(P + \alpha \nabla \tilde{\mathcal{A}}(P))$. Table II describes a distributed version of the Line Search Algorithm with a starting position of $P \in \Omega$. The line search starts with a factor α_{max} which scales the smallest nonzero partial to u_{max}, ensuring all robots with nonzero partial derivatives can move the maximum distance,

$$\alpha_{\mathrm{max}} = \frac{u_{\mathrm{max}}}{\min\{\|\nabla_i \tilde{\mathcal{A}}(P)\| \mid \nabla_i \tilde{\mathcal{A}}(P) \neq 0\}}. \qquad (8)$$

Lemma VI.1 *The*Distributed Line Search Algorithm *is equivalent to the* Line Search Algorithm *with* $F = \tilde{\mathcal{A}}$*.*

```
Name:      DISTRIBUTED LINE SEARCH ALGORITHM
Goal:      Compute step size for gradient descent of $\tilde{\mathcal{A}}$
Input:     Configuration, $P = (p_1, \dots, p_n) \in \mathcal{D}^n$
Assumes:   (i) Connected network of static nodes
           (ii) $S_j$ knows $p_i$, $\nabla_i \tilde{\mathcal{A}}$ and $\Omega_i$ for each robot
           within communication range
           (iii) Step size $\tau$ and tolerance $\theta \in (0,1)$ known a
           priori by all static nodes
Output:    Step size, $\alpha \in \mathbb{R}_{\geq 0}$

Initialization
  $S_1, \dots, S_m$ calculate $\alpha_{\max}$, cf. (8) via a consensus algorithm
For $j \in \{1, \dots, m\}$, node $S_j$ executes concurrently
  $\alpha = \alpha_{\max}$
  repeat
    calculates $\tilde{\mathcal{A}}_j\left(P'_j(\alpha, P)\right) - \tilde{\mathcal{A}}_j(P)$ and $\mathrm{d}_j(\alpha, P)^2$
    execute consensus algorithm to get the following:
      $\tilde{\mathcal{A}}(P'(\alpha, P)) - \tilde{\mathcal{A}}(P) = \sum_{j=1}^{m} \tilde{\mathcal{A}}_j\left(P'_j(\alpha, P)\right) - \tilde{\mathcal{A}}_j(P)$
      $\left\|P - P'(\alpha, P)\right\|^2 = \sum_{j=1}^{m} \mathrm{d}_j(\alpha, P)^2$
    $\nu = \frac{\theta}{\alpha}\left\|P - P'(\alpha, P)\right\| + \tilde{\mathcal{A}}(P'(\alpha, P)) - \tilde{\mathcal{A}}(P)$
    if $\nu > 0$ then
      $\alpha = \alpha\tau$
  until $\nu \leq 0$
```

TABLE II
DISTRIBUTED LINE SEARCH ALGORITHM.

We are ready to present our technique for a greedy optimization algorithm. At timestep k, the nodes follow a gradient descent algorithm to define a sequence of configurations, $\{P^{\dagger}_{\gamma}\}$, $\gamma \in \mathbb{N}$, such that $P^{\dagger}_1 = P^{(k)}$ and

$$P^{\dagger}_{\gamma+1} = \mathrm{proj}_{\Omega}\left(P^{\dagger}_{\gamma} - \alpha \nabla \tilde{\mathcal{A}}(P^{\dagger}_{\gamma})\right), \; \alpha \in \mathbb{R}_{\geq 0},$$

where α is chosen via the DISTRIBUTED LINE SEARCH ALGORITHM. When $|\tilde{\mathcal{A}}(P^{\dagger}_{\gamma+1}) - \tilde{\mathcal{A}}(P^{\dagger}_{\gamma})| = 0$, the algorithm terminates, and the nodes set $P^{(k+1)} = P^{\dagger}_{\gamma+1}$. By the end of this calculation, each node knows the identity of robotic agents that belong to its Voronoi cell at timestep $k+1$. Node S_j transmits $p_i(k+1)$ to robot R_i, which then moves to that location between timesteps. The overall gradient descent algorithm is summarized in Table III.

Proposition VI.2 *The* DISTRIBUTED PROJECTED GRADIENT DESCENT ALGORITHM *is distributed over $\mathcal{N}$. Moreover, under the assumptions of Proposition V.6, any execution is such that*

- *the robots do not collide,*
- *at each timestep after the first, measurements are taken at stationary configurations of $P \mapsto \tilde{\mathcal{A}}^{(k)}(P)$ over $\Omega^{(k)}$.*

Remark VI.3 (Robustness to agent failures) The proposed algorithm is robust to agent failures. If an agent stops sending position information to the nodes, it ceases to receive new control vectors and remains in place. Meanwhile, the rest of the network carries on its operation with the available resources and will eventually take measurements in the areas previously covered by the failing agents. •

B. Simulation results

We performed simulations with the following parameters: $m = 5$ static nodes, $n = 7$ robotic agents, and the domain $\mathcal{D} = \{(0, .1), (2.5, .1), (3.45, 1.6), (3.5, 1.7), (3.45, 1.8), (2.7, 2.2), (1, 2.4), (0.2, 1.3)\}$. We used the separable covariance function defined by $\mathrm{Cov}[Z(s_1, t_1), Z(s_2, t_2)] = C_{\mathrm{tap}}(\|s_1 - s_2\|, 0.4) C_{\mathrm{tap}}(|t_1 - t_2|, 5.5)$, where

$$C_{\mathrm{tap}}(\delta, r) = \begin{cases} e^{-\frac{\delta}{10r}}\left(1 - \frac{3\delta}{2r} + \frac{\delta^3}{2r^3}\right) & \text{if } \delta \leq r, \\ 0 \text{ otherwise.} \end{cases}$$

This is a tapered exponential function belonging to the class of covariance functions suggested in [4]. We used $\omega = \sigma_{\epsilon}^2 = 0.02$, and $u_{\max} = 0.3$.

We compared the performance of our algorithm against two naive data collection strategies, using the actual average error variance $\mathcal{A}^{(k)}$ as a metric. In the first, the robots remained motionless in a configuration such that each robot is located at the incenter of its own Voronoi cell. Next we tried a lawnmower approach. We divided the environment vertically among the robots, and had them march back and forth along horizontal trajectories, avoiding the boundary of the region. Finally, we ran the DISTRIBUTED PROJECTED GRADIENT DESCENT ALGORITHM from the same starting position as the lawnmower approach. Each experiment ran for $k_{\max} = 20$ steps. Agent R_2 stopped transmitting measurements at time $k = 3$, while R_5 stopped at $k = 5$. Figure 2 shows the trajectories traveled by all agents in the lawnmower approach and by two representatives in the gradient descent algorithm. In the latter algorithm, the two agents which stopped sending measurements ceased to move. The other agents avoided colliding with them, but filled in nearby due to the gradient. Figure 3 shows a plot of the errors as k increases from 1 to $k_{\max}$. It can be seen that the gradient algorithm has smaller error than either of the a priori approaches.

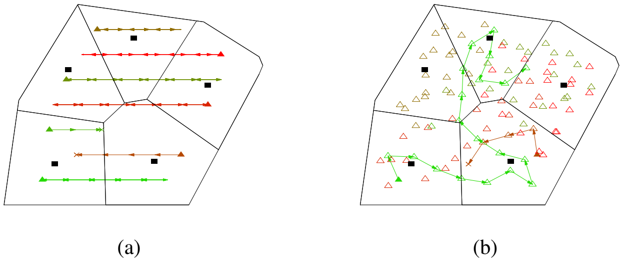

Fig. 2. (a) Trajectories of all robots following the a priori lawnmower path, and (b) two representative trajectories from the distributed projected gradient descent algorithm. The filled squares represent the (static) positions of the nodes, and the filled triangles show the starting positions of the robots. The X's show robots which ceased communicating. The empty triangles in (b) show all measurements taken during the experiment.

VII. CONCLUSIONS AND FUTURE WORK

We have considered a network composed of robotic sensors and static nodes performing spatial estimation tasks. We have focused on the problem of optimizing data acquisition in order to better estimate a spatiotemporal random field. We have used the average error variance of the kriging estimator as a metric for the design of optimal measurement

Name: Distributed Projected Gradient Descent Algorithm

Goal: Find a local minimum of $\tilde{\mathcal{A}}^{(k)}$ within $\Omega^{(k)}$.

Assumes: (i) Connected network of static computing nodes and mobile robotic sensing agents
(ii) Static nodes deployed over $\mathcal{D}$ such that $R \geq \max_{i\in\{1,\dots,m\}} \{\mathrm{CR}(V_i(Q))\} + r + u_{\max}$
(iii) Step size τ and tolerance value $\theta \in (0, 1)$ known a priori by all nodes (iv) Some termination marker known to all nodes and robots which may be sent to mark the end of a gradient descent loop.

Uses: (i) Each node uses the temporary vectors P_{cur}, respectively P_{next} to hold the configuration at the current, respectively next step of the gradient projection algorithm. For ease of exposition, we use global notation although S_j only calculates and uses the parts of these vectors which correspond to agents currently within communication range.

```
At time $k \in \mathbb{Z}_{\geq 0}$, node $S_j$ executes:
sets $R_{\text{cov}}(j) = \{R_i \mid \mathrm{d}(p_i(k), V_j(Q)) \leq r\}$
collects measurements and locations from all robots in $R_{\text{cov}}(j)$
sets $P_{\text{next}} = P^{(k)}$
repeat
    sets $P_{\text{cur}} = P_{\text{next}}(j)$ and calculates $-\nabla\tilde{\mathcal{A}}_j^{(k)}(P_{\text{cur}})$
    transmits vector $\nabla_i\tilde{\mathcal{A}}_j^{(k)}(P_{\text{cur}})$ to all robots in $R_{\text{cov}}(j)$
    collects sum $\nabla_i\tilde{\mathcal{A}}^{(k)}(P_{\text{cur}})$ from all robots in $R_{\text{cov}}(j)$
    runs Distributed Line Search Algorithm at $P_{\text{cur}}$ to get $\alpha$
    sets $P_{\text{next}} = P_{\text{cur}} + \alpha\nabla\tilde{\mathcal{A}}^{(k)}(P_{\text{cur}})$
    calculates $|\tilde{\mathcal{A}}(P_{\text{next}}) - \tilde{\mathcal{A}}(P_{\text{cur}})|$ via consensus
until $|\tilde{\mathcal{A}}(P_{\text{next}}) - \tilde{\mathcal{A}}(P_{\text{cur}})| = 0$
sets $P^{(k+1)} = P_{\text{next}}$
sends a termination marker to all robots currently in $V_j(Q)$
conveys $p_i(k+1)$ to robots that currently belong to $V_j(Q)$
```

```
At time $k \in \mathbb{Z}_{\geq 0}$, robot $R_i$ executes:
sets $S_{\text{cov}}(i) = \{S_j \mid \mathrm{d}(p_i(k), V_j(Q)) \leq r\}$
takes measurement at $p_i(k)$
sends measurement and position to all nodes in $S_{\text{cov}}(i)$
repeat
    receives vectors $\nabla_i\tilde{\mathcal{A}}_j^{(k)}(P^{(k)})$ from all nodes in $S_{\text{cov}}(i)$
    calculates sum $\nabla_i\tilde{\mathcal{A}}^{(k)}(P^{(k)})$
    sends $\nabla_i\tilde{\mathcal{A}}^{(k)}(P^{(k)})$ to all nodes in $S_{\text{cov}}(i)$
until receives termination marker from any node
receives next location $p_i(k+1)$
moves to $p_i(k+1)$.
```

TABLE III
Distributed Projected Gradient Descent Algorithm.

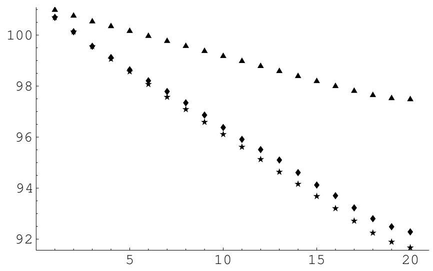

Fig. 3. Average errors up to the kth step of the static (triangle), lawnmower (diamond), and gradient descent (star) approaches.

trajectories of the robots. In our approach, mobile robots take measurements of the environment and static nodes are responsible for collecting the measurements and computing locally optimal configurations for estimation. The design of the overall coordination algorithm combines Voronoi partitions, distributed projected gradient descent, and kriging interpolation technique. We have compared in simulations the performance of our approach against a static network configuration and a lawnmower-based approach.

Future work will focus on: the investigation of theoretical guarantees on the accuracy of the approximation, $\tilde{\mathcal{A}}^{(k)}$, and on the performance and robustness to failure of the proposed coordination algorithm; the development of statistically-sound techniques for the case when, in universal kriging, any particular node only has a small number of measurements available to it; and the quantification of the communication requirements of the proposed approach.

Acknowledgments

Work partially supported by NSF CAREER Award ECS-0546871.

References

[1] M. L. Stein, *Interpolation of Spatial Data. Some Theory for Kriging*. Springer Series in Statistics, New York: Springer Verlag, 1999.

[2] N. A. C. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1993. revised edition.

[3] M. L. Stein, "The screening effect in kriging," *The Annals of Statistics*, vol. 30, pp. 298–323, Feb. 2002.

[4] R. Furrer, M. G. Genton, and D. Nychka, "Covariance tapering for interpolation of large spatial datasets," *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, pp. 502–523, 2006.

[5] F. Pukelsheim, *Optimal Design of Experiments*, vol. 50 of *Classics in Applied Mathematics*. Philadelphia, PA: SIAM, 2006.

[6] E. P. Liski, N. K. Mandal, K. R. Shah, and B. K. Sinha, *Topics in Optimal Design*, vol. 163 of *Lecture Notes in Statistics*. New York: Springer Verlag, 2002.

[7] H. Gao, J. Wang, and P. Zhao, "The updated kriging variance and optimal sample design," *Mathematical Geology*, vol. 28, no. 3, pp. 295–313, 1996.

[8] J. S. Willcox, J. G. Bellingham, Y. Zhang, and A. B. Baggeroer, "Performance metrics for oceanographic surveys with autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 711–725, 2001.

[9] D. O. Popa, K. Sreenath, and F. L. Lewis, "Robotic deployment for environmental sampling applications," in *International Conference on Control and Automation*, (Budapest, Hungary), pp. 197–202, June 2005.

[10] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 45, pp. 48–74, Jan. 2007. Special Issue on Networked Control Systems.

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

[12] S. Martínez, "Distributed representation of spatial fields through an adaptive interpolation scheme," in *American Control Conference*, (New York City, NY), pp. 2750–2755, 2007.

[13] F. Zhang, E. Fiorelli, and N. E. Leonard, "Exploring scalar fields using multiple sensor platforms: tracking level curves," in *IEEE Conf. on Decision and Control*, (New Orleans, LA), pp. 3579–3584, Dec. 2007.

[14] D. P. Bertsekas, "On the Goldstein-Levitin-Polyak gradient projection method," *IEEE Transactions on Automatic Control*, vol. 21, no. 2, pp. 174–184, 1976.