A

In-Situ Soil Moisture Sensing: Optimal Sensor Placement and Field Estimation

XIAOPEI WU, University of Electronic Science and Technology of China, University of Michigan
MINGYAN LIU, University of Michigan
YUE WU, University of Electronic Science and Technology of China

We study the problem of optimal sensor placement in the context of soil moisture sensing. We show that the soil moisture data possesses some unique features that can be used together with the commonly used Gaussian assumption to construct more scalable, robust and better performing placement algorithms. Specifically, there exists a coarse-grained monotonic ordering of locations in their soil moisture level over time, both in terms of its first and second moments, a feature much more stable than the soil moisture process itself at these locations. This motivates a clustered sensor placement scheme, where locations are classified into clusters based on the ordering of the mean, with the number of sensors placed in each cluster determined by the ordering of the variances. We show that under idealized conditions the greedy mutual information maximization algorithm applied globally is equivalent to that applied cluster by cluster, but the latter has the advantage of being more scalable. Extensive numerical experiments are performed on a set of 3-dimensional soil moisture data generated by a state-of-the-art soil moisture simulator. Our results show that our clustering approach outperforms applying the same algorithms globally, and is very robust to lack of training and errors in training data.

Categories and Subject Descriptors: F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms: Measurement, Design

Additional Key Words and Phrases: Soil moisture, 2D/3D sensor placement, Gaussian process, Gaussian regression, Coarse-grained orderings

ACM Reference Format:
X.Wu, M. Liu and Y.Wu. 2011. In-Situ Soil Moisture Sensing: Optimal Sensor Placement and Field Estimation. ACM Trans. Sensor Netw. V, N, Article A (January YYYY), 30 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Soil moisture measurement has many applications in hydrology and is one of the most important indicators in agricultural drought monitoring. For instance, it is a measurement need in four out of the six [1] strategic focus area roadmaps (climate, carbon, weather, and water roadmaps) put forward by NASA as part of its Earth Science mission [NASA 2006]. Specifically, soil moisture measurements are used in all land sur-

[1]The six areas are climate, carbon, surface, atmosphere, weather, and water.

Author's addresses: X. Wu and M. Liu, Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109-2122; Y. Wu, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

face models, all water and energy balance models, general circulation models, weather prediction models, and ecosystem process simulation models.

Depending on the specific application needs, soil moisture may need to be measured with different sampling characteristics. Traditionally, soil moisture on a large scale has been measured by remote radar sensing techniques, e.g., the NASA Soil Moisture Active and Passive (SMAP) mission [SMAP 2008], which uses low-frequency microwave radar and radiometer to sense surface moisture conditions over global land surfaces. Unfortunately, remote sensing such as SMAP has a very large footprint (on the order of square kilometers), and the resulting measurement is only a coarse-resolution representation of a field mean [Martinez-Fernandez and Ceballos 2005; Cosh et al. 2004] over this large footprint. In order to obtain finer-resolution measurements, one has to use in-situ sensing techniques, e.g., in the form of moisture probes buried under the ground. In-situ sensing data has many uses, one of which is the validation and calibration of remote sensing measurements.

Recent advances in integrated sensing and wireless communication have made in-situ sensor networks a reality in a variety of applications, ranging from battle-field surveillance to habitat monitoring to environment observation and forecast systems, and so on [Akyildiz et al. 2002]. Our prior work [Moghaddam et al. 2010] has developed sensor web capabilities to enable flexible and guided sampling strategies for such in-situ sensors.

The main limitation of in-situ sensing is its lack of scalability: since moisture probes need to be buried under the ground at various pre-determined depths up to 2 meters deep, it is impractical and cost-prohibitive to deploy them at high densities over a large area. Note that the cost barrier has as much to do with the cost of installation as with the cost of moisture probes themselves[2]. There exist other environmental sensing scenarios (e.g., ambient temperature, pressure), where large quantities of inexpensive sensors may be rapidly deployed at very high densities. Clearly in soil moisture sensing we cannot afford to do so, and consequently careful consideration has to be given to the sensor placement problem: for a given sensing field, how many sensors are needed and where to place them so as to achieve the best cost-benefit tradeoff.

This sensor placement problem is fundamental to many sensing applications involving the monitoring of spatial phenomena, and it induces an associated problem of field estimation: how to estimate values at locations we don't directly observe (i.e., where no sensors are placed; these will also be referred to as *unobserved locations* throughout the paper) using observations at locations with sensors (these will also be referred to as *observed locations*).

The above problem has received much attention in recent studies. It can be more generally framed as model-based classical experimental design or subset selection problem (see e.g., [Byers and Nasser 2000; Bian et al. 2006; Das and Kempe 2008b]). In the statistics community, classical and Bayesian experimental design focuses on the question of selecting observations to maximize the quality of parameter estimates in linear models (e.g., [Atkinson 1988; Lindley 1956]). One widely used model assumes the spatial statistics of real-world phenomena is Gaussian, i.e., the underlying spatial phenomenon follows a Gaussian distribution. Under such a Gaussian model, information-theoretic measures, notably entropy (e.g. [Ko et al. 1995; Kemppainen et al. 2008;

[2]The Decagon EC-5 moisture sensor probes we used in our deployment cost about $60 each. We typically deploy 3-4 vertically at each location. The mounting (of a wireless transceiver module and actuator) solution involves 10-foot-tall landscape posts, PVC pipes, totaling $40 per location. The installation is very labor intensive, including location marking (done by foot and tape measure over a square kilometer area), special equipment rental (tractor with hole-digging attachment at the back), skilled labor to operate the machinery (digging a 4-foot deep hole 6-inch in diameter per location.

Caselton and Husain 1980; Caselton and Zidek 1984]) and mutual information (e.g. [Guestrin et al. 2005; Krause et al. 2007]) have been frequently used (mutual information is also used in many other settings including sensor query [Ertin et al. 2003] and adaptive sampling [Choi 2009]). The former aims at minimizing the uncertainty in the prediction, after the observations are made, while the latter tries to maximize the information contained in the observation about unobserved locations. All the criteria/objectives mentioned above yield challenging combinatorial optimizations problems. As a result, heuristic algorithms (such as greedy algorithms) have been widely exploited. More detailed discussion on this is given in Section 2.2.

There have also been studies on sensor placement for the purpose of providing coverage, rather than field estimation (see e.g., [Hochbaum and Maas 1985; González-Banos 2001]). Typically in such studies a sensor is assumed to have a certain (disk) coverage area, and a common objective is to place the sensors in such a way that the total area not covered is minimized. Both the objectives and the methodologies used in these studies are very different from what is investigated in the present paper.

In this paper we study the sensor placement problem specifically for the application of soil moisture sensing. Our primary goals are scalability, robustness, and performance. Since the deployment field size for soil moisture sensing is typically on the order of square kilometers[3], depending on the heterogeneity of the soil types and vegetation covers we may be facing tens of thousands of possible locations from which to choose a few tens or a few hundreds for placement. Even as an offline procedure this can be a computationally prohibitive exercise if the placement algorithm is not scalable. This problem is further exacerbated if we wish to jointly design placement and sensor measurement scheduling policies [Shuman et al. 2010]. Secondly, these sensors are intended to operate for long periods of time (months to years), and their observations are used to provide estimates over long periods of time in which the soil moisture process goes through dynamic changes that may or may not be stationary. As a result, the placement decision needs to be highly robust against lack of training and any errors in prior information to produce good estimation performance.

We show that soil moisture data possesses some unique features that can be exploited to achieve the above goals. Specifically, the dynamics in soil moisture content greatly depend on factors such as soil type and vegetation cover, which are slow-changing over time. These stationary features predict very reliably the *relative* moisture levels between two different locations, even if the *absolute* moisture values are constantly changing. As a result, there exists a coarse-grained monotonic ordering of locations in terms of their soil moisture levels over time, a feature much more stable than the soil moisture process itself at these locations. This feature leads us to consider a particularly simple and highly scalable *clustered* sensor placement scheme, where locations are classified into clusters based on this coarse-grained ordering. Furthermore, we show that the variances of soil moisture levels at different locations also obey this coarse-grained ordering quite well. This feature is then used to determine how many sensors to place within each cluster, given a total budget. With this two-step approach, the overall placement problem is divided into separate problems within each cluster. This scheme can be easily combined with any existing sensor placement algorithm, such as those mentioned above.

The key idea underlying this approach is to group statistically similar locations together. We formally justify this statistical clustering idea and show under what conditions it produces equivalent result as a global placement approach. We further show that this clustering scheme may be viewed as a much simpler and highly effective

[3]This is because in order to use in-situ sensing to calibrate remote sensing measurement (or numerical simulation tools), we need the size to be on the same order.

way of approximating the well-known *K-means clustering* algorithm [Jain et al. 1999; MacQueen 1967; D.Vinod 1969]. We conduct extensive numerical experiments using a large set of 3-dimensional soil moisture data generated by a state-of-the-art soil moisture simulator, the TIN-based Real-time Integrated Basin Simulator (tRIBS) [Vivoni et al. 2005; Flores et al. 2009]. We evaluate and compare different sensor placement and field estimation algorithms. We conclude that the coarse-grained ordering of locations is a far more stable feature inherent in the soil moisture data that leads to much more scalable and robust placement algorithms. In addition, these algorithms in general outperform their global counterparts which only rely on the Gaussian assumption.

The rest of this paper is organized as follows. In Section 2 we formally define the sensor placement problem and review existing methods in Section 2.2. We analyze basic statistical features of soil moisture data in Section 3. We then exploit a coarse-grained ordering to propose a clustering approach to the sensor placement problem, and analyze its performance and properties in Section 4. Numerical results are presented in Section 5 and practical application of these placement algorithms are discussed in Section 6. We conclude this paper in Section 7.

2. PROBLEM DESCRIPTION AND PRELIMINARIES

In this section we first define the sensor placement and field estimation problem, and then review commonly used placement schemes.

2.1. The sensor placement problem

For a given field of interest, the (discrete) sensor placement problem is stated as follows. There is a set of possible locations, denoted by V, $|V| = N$, where we could place sensors to take in-situ or point measurements. Note that these locations do not have to be confined within a 2D plane; the exact same formulation applies to higher dimensional placement problems. For locations where we place a sensor, we obtain perfect observations of the underlying phenomenon, often described as a spatial random process; for those where we do not place sensors, we have to provide an estimate based on direct observations elsewhere. The objective is to select a subset $A \subset V$, $|A| = K$, to place sensors so as to minimize a certain measure of the estimation error over unobserved locations.

More formally, let $V = [v_1, v_2, \cdots, v_N]$ denote the set of N locations in the field, and $X_V = [X_1, X_2, \cdots, X_N]$ the corresponding random variables describing the observations at these locations. For any subset $A \subset V$, we will similarly use X_A to denote the collection of random variables associated with the set of locations A. The optimal sensor placement problem (P) is given as follows:

$$\textbf{(P):} \qquad A^* = \arg \min_{A \subset V, |A|=K} \mathbb{E}\left[err(X_V, \hat{X}_V(A)) \right] , \tag{1}$$

where $\hat{X}_V(A)$ is the optimal estimate given direct observations made at locations A, $err()$ is a certain error function measuring the distance between the real value and the estimate, $\mathbb{E}[.]$ denotes expectation, and it is over the joint distribution of the random vector X_V. We will assume that sensor measurements are noiseless, in which case problem (P) can be rewritten as

$$\textbf{(P):} \qquad A^* = \arg \min_{A \subset V, |A|=K} \mathbb{E}\left[err(X_{V \setminus A}, \hat{X}_{V \setminus A}(A)) \right] , \tag{2}$$

where $V \setminus A$ denotes the set of all elements in V but not in A. A typically used error function is the mean-squared error (MSE), also used in this paper:

$$err(X_{V\setminus A}, \hat{X}_{V\setminus A}(A)) = \|\hat{X}_{V\setminus A}(A) - X_{V\setminus A}\|^2 .$$

To solve this problem we would need to know the distribution of the random process, which is often approximated in practical applications. Even when this distribution is precisely known, the above problem remains a very difficult one due to the combinatorial nature of the location subset selection, as well as the fact that it is a joint optimization of subset selection and the selection of the best estimate for a given error function.

To address these difficulties, a commonly used approach is to simplify the second problem, which is the selection of the best estimate for a given error function. For instance, one can limit the solution space of the estimation problem to the set of linear estimates, as is done in [Das and Kempe 2008a], and then solve the joint optimization problem. An even more commonly used approach is to simply assume that the underlying field is described by a Gaussian random process; this is described next. Algorithms based on this assumption are commonly used for sensing temperature and humidity (see e.g. [Ko et al. 1995; Guestrin et al. 2005; Yang and Blum 2008; Krause et al. 2007]).

2.2. Field estimation and sensor placement based on the Gaussian assumption

Under the Gaussian assumption, any subset of the locations are described by a joint Gaussian distribution. A single most significant property that follows is that the conditional distribution of random variable X_v for some location v, given observations obtained at any subset of locations $A \subset V$ such that $v \notin A$, remains Gaussian, i.e.,

$$X_v|X_A \sim \mathcal{N}(u_v + \Sigma_{vA}\Sigma_{AA}^{-1}(x_A - u_A), \Sigma_{v,v} - \Sigma_{vA}\Sigma_{AA}^{-1}\Sigma_{vA}^T), \tag{3}$$

where x_A denotes the vector of observed values at locations A, u_A the mean vector, and Σ_{AA} the covariance matrix of the random variables X_A. $\Sigma_{vA}(= \Sigma_{Av}^T)$ is a covariance vector with elements $\{\Sigma_{VV}\}_{v,w}$, $\forall w \in A$, the covariance between random variables X_v and X_w. For convenience, below we will also use $\Sigma_{v,w}$ to denote the covariance between random variables X_v and X_w.

Therefore, if sensors are placed only at locations in the subset A, and the observations x_A are made, then the best estimate for the (unknown) observation at location $v(v \in V\setminus A)$ is the mean of this conditional distribution:

$$u_{v|A} = u_v + \Sigma_{vA}\Sigma_{AA}^{-1}(x_A - u_A) , \tag{4}$$

and the uncertainty is captured by the associated conditional variance:

$$\sigma_{v|A}^2 = \Sigma_{v,v} - \Sigma_{vA}\Sigma_{AA}^{-1}\Sigma_{vA}^T . \tag{5}$$

In practice, when precise knowledge of the underlying distribution is hard to obtain, the above field estimation for a given sensor placement A is typically done in the following two steps: (1) use a set of training data that contains measurements (or simulated quantities) at *all* locations of interest to obtain the empirical mean and covariances (i.e., vector u_V and matrix Σ_{VV}) across all locations; (2) use observations made at locations A to estimate quantities at unobserved locations $V\setminus A$ using Equation (4). This will be referred to as the Gaussian regression (GR) method. Note that GR is generalization of linear regression.

Using this approach the resulting MSE is given by

$$MSE(A) = trace\{\Sigma_{VV} - \Sigma_{VA}\Sigma_{AA}^{-1}\Sigma_{AV}\} . \tag{6}$$

Often a heuristic greedy algorithm is used to obtain a sub-optimal solution. It operates sequentially: starting from an empty set $A = \emptyset$, at each time it adds one location v to this set so as to minimize $MSE(A \cup v)$. In addition to minimizing MSE, there are two other commonly used criteria based on the notions of entropy and mutual information as an indirect way of measuring the MSE. These are described next.

One way to reduce estimation error is to minimize the uncertainty of the unobserved locations ($V \setminus A$) given the selection of A, or their entropy:

$$\begin{aligned} \textbf{(P1):} \quad A^* &= \arg \min_{A \subset V, |A|=K} H(X_{V \setminus A} | X_A) \\ &= \arg \min_{A \subset V, |A|=K} H(X_V) - H(X_A) \\ &= \arg \max_{A \subset V, |A|=K} H(X_A) \ . \end{aligned}$$

With the Gaussian assumption, $H(X_A)$ can be expressed in closed form and obtained relatively easily. Specifically, the entropy of a Gaussian random variable $X_w (w \in V \setminus A)$ conditioned on a set of random variables X_A is given by

$$H(X_w | X_A) = \frac{1}{2} \log(2\pi \sigma^2_{w|A}) \ . \tag{7}$$

If $A = \{w_1, \cdots, w_K\}$, then $H(X_A)$ is obtained by using the chain-rule:

$$\begin{aligned} H(X_A) = H(X_{w_K} | X_{A \setminus w_K}) + H(X_{w_K - 1} | X_{A \setminus \{w_K, w_K - 1\}}) + \cdots \\ + H(X_{w_2} | X_{w_1}) + H(X_{w_1}) \ . \end{aligned} \tag{8}$$

Finding the subset with the largest entropy remains a combinatorial problem. It was shown in [Ko et al. 1995] that this problem is NP-complete. For this reason, a greedy suboptimal algorithm was proposed in [Ko et al. 1995], whereby the sensor placement is done in a sequential manner: each time a single location with the highest conditional entropy, conditioned on the locations chosen in previous steps, is selected.

Typically the highest entropy set contains locations that are as far as possible from each other. The result is that they will tend to be on the boundary. An alternative is thus introduced to address this limitation, by using mutual information as a measure of the usefulness of observations at selected locations [Guestrin et al. 2005; Krause et al. 2007]. With this selection criterion, the problem becomes to seek a subset of locations such that the mutual information between the selected and the remaining locations is maximized:

$$\begin{aligned} \textbf{(P2):} \quad A^* &= \arg \max_{A \subset V, |A|=K} MI(X_{V \setminus A}, X_A) \\ &= \arg \max_{A \subset V, |A|=K} H(X_{V \setminus A}) - H(X_{V \setminus A} | X_A) \ . \end{aligned}$$

Under the Gaussian assumption, $H(X_{V \setminus A})$ can be obtained from Equation (8), and $H(X_{V \setminus A} | X_A)$ is given by:

$$H(X_{V \setminus A} | X_A) = \frac{1}{2} log((2\pi e)^N det(\Sigma_{V \setminus A | A}) \ ,$$

where $\Sigma_{V \setminus A | A}$ is the predictive covariance matrix, which can be inferred using Equation (5). However, for the same reason mentioned earlier, problem (P2) also remains NP-hard. A similar sequential greedy algorithm is proposed in [Guestrin et al. 2005; Krause et al. 2007], where in each step a location is chosen if its addition to the set of

locations already selected results in the largest mutual information with the remaining set of locations. This algorithm is shown to have a constant approximation ratio with respect to the optimal solution of (P2), when $K \ll N$, using submodularity.

The above two problems will be referred to as MaxEN and MaxMI, respectively, in our subsequent discussion, and the original problem will be referred to as MinMSE. Note that due to the Gaussian assumption the optimization in all three problems is independent of the observation data and thus can be carried out offline. Furthermore, the resulting sensor placement is deterministic in all cases and the corresponding greedy algorithms work similarly.

While these greedy algorithms mentioned above are conceptually very easy to implement, the associated computational cost is quite high, especially for large N and K. This is because the greedy updates for both EN and MI require the computation of conditional entropy using Equation (8), which involves solving a system of $|A|$ linear equations. As noted in [Krause et al. 2007], the computational complexity of greedily updating EN and MI is $O(K^3)$ and $O(KN^4)$, respectively.

3. SOIL MOISTURE DATA

In this section we describe the soil moisture data set used in this paper. The same data set is also used for numerical experiments presented in Section 5.

3.1. Simulated soil moisture data

There is a lack of large scale real soil moisture measurement data existing in the literature to the best of our knowledge. This is partly because soil moisture has traditionally been measured using remote sensing methods like radar and radiometers that, as mentioned earlier, typically have relatively large footprints (at least on the order of square kilometers). The resulting measurements are coarse-grained. Fine-grained in-situ sensing has only recently become available following advances in wireless sensor network technologies. However, even in this case collecting real data can be a very labor intensive and costly exercise, as one typically needs to bury multiple (e.g., 8-10) moisture probes under the ground at different depths, up to 1 or 2 meters deep, at each sensing location. For these reasons large scale field collection of soil moisture data is still an area in its early stage[4].

For lack of real data, in this study we will instead use simulated soil moisture data generated by a state-of-the-art soil moisture simulator, called the TIN-based Real-time Integrated Basin Simulator (tRIBS) [Vivoni et al. 2005; Flores et al. 2009]. Soil moisture varies as a function of time and three-dimensional (3D) space in response to variable exogenous forcings such as rainfall, temperature, cloud cover, and solar radiation. It is also influenced by landscape parameters such as vegetation cover, soil type, and topography. The soil moisture variations in time and depth, or infiltration, can be modeled as a pair of partial differential equations (the so-called Richards equations) in the case of a flat horizontal surface.

For a homogeneous and flat landscape, the spatial variations can be assumed to be limited to one dimension or 1D (depth only). In the presence of topography, the 1D infiltration in the direction of the surface normal is redistributed, dominated by gravity, by the lateral fluxes in the vadose (unsaturated) zone as well as the boundary values imposed by the phreatic (saturated) zone at depth. To efficiently model local topography for the purpose of this lateral redistribution process, triangulated irregular network (TIN) surface models can be used, which discretize the surface topography

[4]In a separate but related effort, we are in the process of collecting large scale fine-grained in-situ soil moisture data that could eventually be used for a variety of purposes, including studying sensor placement. The system architecture is described in [Moghaddam et al. 2010].

into triangle-shaped mesh elements (also called voronoi cells) for subsequent numerical analysis. Proper modeling of the soil moisture evolution process has to take into account the water flow mechanisms and the energy balance of the entire landscape, including the surface-atmosphere interactions. It therefore has to include mechanisms such as rainfall, groundwater flow, evapotranspiration demand, and runoff. Among the most sophisticated numerical models capable of predicting the time-space soil moisture evolutions is the TIN-based Real-time Integrated Basin Simulator (tRIBS) [Vivoni et al. 2005; Flores et al. 2009]. The mesh-generation algorithm within tRIBS is an adaptive discretization scheme that resembles the spatial pattern of the landscape with variable resolution to ensure that the impact of the basin response is properly represented.

This model is used to simulate long time-series realizations of the soil-moisture over a 2km x 2km basin with an arbitrary topography and drainage channels, which is shown in Figure 1(a)[5]. Soil moisture observations are collected at 9 different depths at 2400 different locations/points over this sensing field. The entire data set consists of 2208 snapshots taken over a period of three months (in simulation time), once per hour. Each snapshot thus contains 2400 observation vectors; each location produces a vector of readings at 9 depths. Figure 1(b) shows an example of such a snapshot.

3.2. Statistical Characterization of Soil Moisture

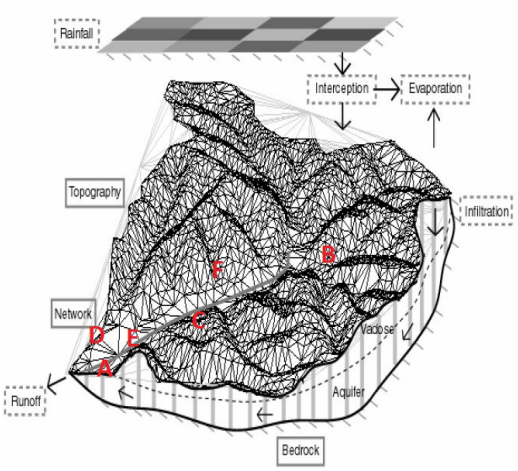
There have been a number of studies on statistical characterization of soil moisture variability. Previous works have generally recommended either a Gaussian or Beta distribution to represent surface soil moisture variations [Bell et al. 1980; Francis et al. 1986; Hawley et al. 1983; Hills and Reynolds 1969; Wilson et al. 2003; Famiglietti et al. 1999; Western et al. 2002]. We note that in order to be more realistic, one must take into account the fact that the evolution of soil moisture in any environment (of reasonable size) is often affected by very high heterogeneity, e.g., in precipitation, soil type, topography, and vegetation cover. Consequently, the statistical characterizations proposed in the above studies are often gross simplifications and do not describe well more realistic data.

In Figure 2(a) we show the soil moisture[6] observed at 3 (out of 2400) randomly selected locations (at the depth closest to surface, also referred to as the surface soil moisture) over the three-month period. The figures at the top show the histograms of each location as well as their corresponding Gaussian kernel density estimates, respectively. The figures at the bottom are the estimated probability density functions (PDFs) of each, respectively. It can be clearly seen that the data does not following a Gaussian distribution; indeed in all three cases the data exhibits a multimodal behavior (here a mode refers to a local maximum in the density function [Efron and Tibshirani 1993]). For completeness, we also show the temporal variations at these locations over the three months in Figure 2(b). While only three locations are shown here due to space limit, we note that this is a general observation drawn across all locations in our data set. In Figure 3 we further show the joint distribution between a pair of locations. We again note the multiple peaks in these empirical distributions.

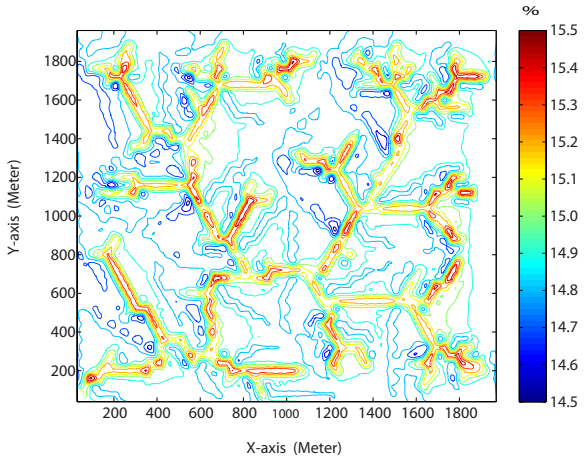
The above observation suggests that while we may still use sensor placement and field estimation methods derived based on the Gaussian assumption (note that the Gaussian assumption is used in (1) the computation of entropy and mutual information, and (2) the estimation using conditional expectation, both of which become approximations if the Gaussian assumption does not hold), we should also seek other

[5]Figure courtesy of D. Entekhabi and A. Flores.

[6]For ease of presentation, the soil moisture shown here as well as that used in our numerical studies have been enlarged 1000 times over the actual soil moisture readings.

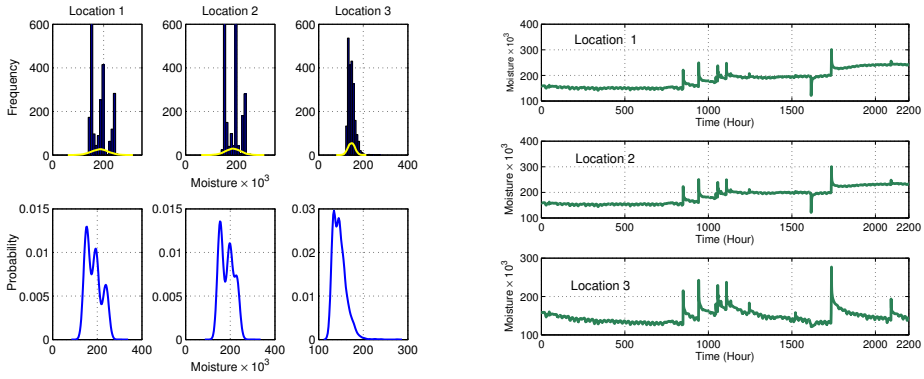


(a)



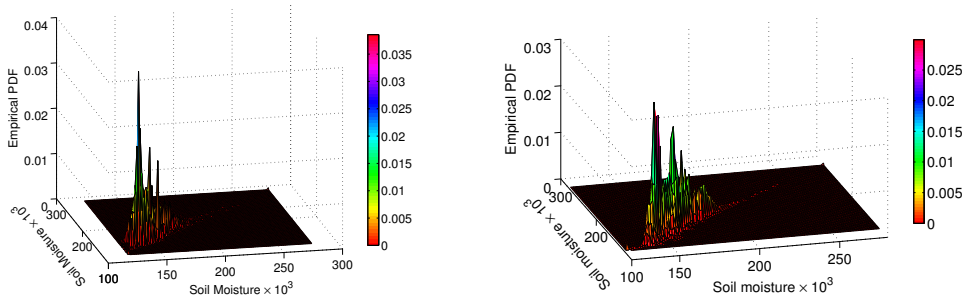
(b)

Fig. 1. (a) A nominal 2km x 2km basin used for tRIBS simulations. This example is assumed to be climatologically consistent with Oklahoma. (b) A sample snapshot of the soil moisture readings (in %) over the terrain of (a) generated by tRIBS.



(a) (b)

Fig. 2. (a) (Top) Frequency histograms of soil moisture at three locations. (Bottom) Estimated probability density functions at three locations. (b) Temporal changes at these locations overtime.



(a) (b)

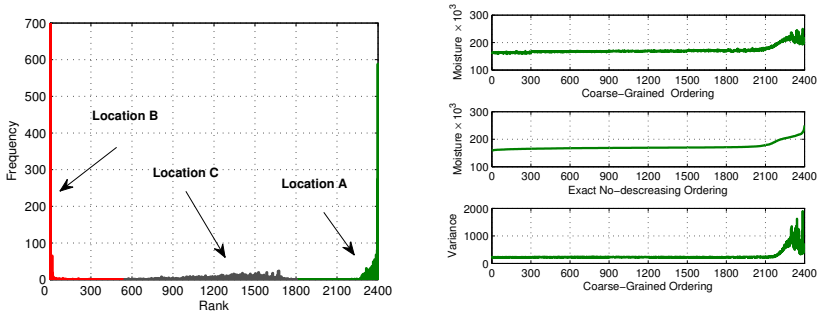
Fig. 3. Two-dimensional joint probability distribution of soil moisture at locations (a) (990,1000) and (50,80), and (b) (1690,1680) and (1690,1040), in the simulated filed.

features embedded in the data that may enhance the performance and help us in constructing good sensor placement algorithms.

4. CLUSTERING BASED ON A COARSE-GRAINED MONOTONIC ORDERING

In this section we examine some interesting features exhibited by this data, which in turn motivate two key ideas explored in our sensor placement algorithm. This is followed by an analysis of the validity of the key ideas and how they relate to existing literature. We end this section by discussing the applicability of alternative methods.

As mentioned in the introduction, a key consideration in this study is the scalability of the placement algorithm. Unlike indoor sensing applications, the size of soil moisture sensing field needed to calibrate the remote sensing measurement or numerical tools is usually on the order of a few square kilometers. This means the placement algorithm has to select 10s or 100s of points out of tens of thousands of candidate locations (at the resolution of 10x10 square meters over a field of a few square kilometers). However, none of the placement algorithms cited earlier, with the exception of MaxEN, is particularly scalable (e.g., the most scalable version of MaxMI has a complexity of $O(KN^3)$ [Krause et al. 2007]). One natural idea of improving the scalability for large



(a) (b)

Fig. 4. (a) Rank distribution of three locations (A, B, C labeled on Fig. 1(a)) over all snapshots. (b) Moisture readings under different orderings.

sensing field is to decompose the original placement problem into several, separate smaller-scale problems, and then apply any of the known algorithms to the smaller-scale problems. It turns out that there is a particularly intuitive and simple way of doing so for soil moisture data as we show below.

4.1. Coarse-Grained Monotonic Ordering over Time

It is well understood that the dynamic change in soil moisture greatly depends on factors such as the soil type, vegetation cover, and precipitation to name a few. Indeed it is such understanding in quantitative form that constitutes the computational engine behind the tRIBS simulation tool. In particular, these factors collectively determine the dissipation process of moisture content. It follows that locations with similar surroundings sharing similarity in these factors will exhibit similar dynamics (or the processes at these locations will act similarly over their input). A key observation here is that many such determining factors are relatively stationary over time, including soil type and vegetation cover. These stationary features may predict very reliably the *relative* moisture levels between two different locations, even if the *absolute* moisture values are constantly changing. For instance, at locations near waterways with heavy vegetation we may expect to see consistently higher moisture level than those further away, uphill, and/or amidst sandy soil, regardless of the weather condition. This suggests that there may be a relatively stable ordering of locations in terms of their soil moisture content over time, a feature possibly much more stable than the soil moisture process itself at these locations.

To verify this intuition, we sort the 2400 locations in each of the 2208 snapshots in increasing order of their soil moisture readings. The histograms of the numerical rankings of three locations (labeled A, B and C on Figure 1(a)) are shown in Figure 4(a). We can see that a location's appearances in these sorted sequences are highly concentrated in a certain region (lower $1/8$ for B, middle $1/3$ for C, and top $1/8$ for A), suggesting an underlying soil/location type that enables relatively rapid and slow water dissipation, respectively. These results show that indeed the relative moisture content of a location is fairly steady over time.

Note however, that the ordered sequences are not exactly the same from time to time. In other words, even though a location's appearances in these ordered sequences are relatively steady on a macroscopic level, they are subject to (minor) fluctuations at a microscopic level (e.g., it may rank the 22nd in one snapshot and the 25th in another; occasionally this fluctuation may be even bigger). This motivates us to adopt a coarse-grained sorting method, by loosely classifying the field's all 2400 locations into a few

subsets/clusters depending on their relative soil moisture readings (e.g., they may be interpreted as from "very dry" to "very wet"). Specifically, suppose we evenly divide the set of 2400 numerical rankings into C clusters, and record in which cluster each location falls into in each of the 2208 sorted sequences. We then assign a location to the cluster in which it has appeared the most often (with ties broken randomly). To keep this classification coarse-grained, we will assign a random order to locations that fall within the same cluster. This process results in a particular rank ordering of locations relabeled from 1 to 2400. On the top of Figure 4(b), we show the soil moisture readings at locations ordered this way, from a randomly chosen snapshot, with $C = 8$ clusters. The corresponding moisture variance associated with each location thus ordered is displayed at the bottom of 4(b). To compare, in the middle of Figure 4(b) we show for the same snapshot the exact increasing-ordered sequence of soil moisture readings. As can be seen, the macroscopic ordering of moisture readings are fairly well preserved even though the locations are randomly ordered within each cluster, with the exception of the last cluster (the highest moisture content).

We summarize the main observations as follows. Firstly, the rank ordering of locations according to their soil moisture readings at any given time is fairly consistent over time (i.e., a relatively wet location at some time will likely remain relatively wet at other times). This suggests a very simple way of grouping the locations into clusters of similar moisture levels. Secondly, the rank ordering of locations according to the variance in their soil moisture is fairly consistent with the rank ordering according to the soil moisture values. This suggests a way of determining how many sensors to place within each clusters. These ideas are detailed in the next subsection.

We end this subsection by noting that the above observation is ultimately constrained by the data from which it is derived, which is a 4 km^2 region that is climatologically consistent with Oklahoma[7]. Thus the condition under which this observation holds is that all dynamic factors such as rain fall, air temperature, winds, are consistent with what one sees within such a region (e.g., the spatial continuity in the rainfall process may mean that two locations 20 meters apart received similar amounts (not always, but on average)). This condition is however not an overly limiting one because in practice we will not try to solve a placement problem over an area bigger than this scale - this is roughly the scale of remote sensing so it is sufficient for the purpose of validation and calibration.

4.2. Clustered Sensor Placement Using the Coarse-Grained Ordering

We now introduce the following clustering method to solve the sensor placement problem. It has two elements (or steps). The first is *location clustering*: dividing the entire set of locations into groups/clusters. The second is *sensor allocation*: determining how many sensors to be placed within each cluster given a total budget. With these two steps, we decompose the original sensor placement problem into separate, smaller sensor placement problems. Within each cluster different placement algorithms may be applied.

The location clustering step is accomplished following the description in the previous section. Specifically, we evenly divide the 2400 numerical rankings into C groups, i.e., group i contains the rankings $[(i-1)\frac{2400}{C}+1, i\frac{2400}{C}]$, $i = 1, 2, \cdots, C$, assuming 2400 can be exactly divided by C. We then obtain the ranking distribution of a location v, and count the number of times its rank falls within group i, denoted by $r_v(i)$. Note that $\sum_i r_v(i) = 2208$ as each location has one rank per snapshot. We assign location v to cluster i if $r_v(i) \geq r_v(j)$, $\forall j = 1, 2, \cdots, C$, with ties broken randomly. This process is then repeated till all locations v has been assigned. The output of this step is C

[7]Oklahoma is one of the few geologically typical regions in North America designated by NASA.

clusters denoted by $V_1, V_2, \cdots, V_C$, each of size N_i, $i = 1, 2, \cdots, C$, and $\sum_i N_i = N$. Note that even though the clusters of numerical rankings are equal in size, not all clusters will end up with the exact same number of locations.

The sensor allocation step is done following the principle that clusters with higher variability should be assigned more sensors. As we have seen in the top figure of Figure 4(b) for the first few clusters the variation among observations at different locations is relatively small. This suggests that within these clusters observations are highly correlated, and therefore a small number of sensors might suffice. Specifically, assume that we are interested in placing a total K sensors, with K_i allocated to the ith cluster, $i = 1, 2, \cdots, C$. Denote the mean at location v by μ_v, and the sample mean of cluster i by $\bar{\mu}_i = \frac{1}{N_i} \sum_{v \in V_i} \mu_v$. Then the sample variance of the mean values within cluster i is given by $\bar{\sigma}_i^2 = \frac{1}{N_i - 1} \sum_{v \in V_i} (\mu_v - \bar{\mu}_i)^2$.

The allocation K_i to cluster V_i is set to be proportional to the sample variance of this cluster, and given by:

$$K_i = \frac{\bar{\sigma}_i^2}{\sum_{i=1}^{C} \bar{\sigma}_i^2} K \ . \tag{9}$$

The reason we choose to generate clusters by *evenly* dividing the rankings is because the same placement outcome can be realized through different combinations of the cluster step and the allocation step given the same monotonic ordering. That is, the effect of changing the size of clusters can also be achieved by changing the allocation. For this reason we decided to keep the clustering even, while letting the allocation be determined by the variability of the mean within each cluster, which in general results in uneven sensor allocation. We could also do the opposite thing: to keep the allocation even and determine accordingly the right size of each cluster. In the next subsection as well as in numerical results we compare our approach with an alternative clustering approach, whereby the clustering is determined by an optimization procedure so the result is in general uneven. This is followed by the same allocation method determined by the variability of the mean in each cluster. Numerical results will show that these two are very similar in performance, indicating that the uneven sensor allocation sufficiently compensates for the even division of clusters.

We will examine the use of different placement schemes within this clustering framework, including using MaxEN, MaxMI and MinMSE individually within each cluster. These will subsequently be referred to as *clustered* placements, while their non-cluster based counterparts will be referred to as *global* placement schemes.

It's worth noting that the above clustering method is based on statistical similarities among locations, inferred from their coarse-grained ordering. This decomposition not only guarantees observations in each cluster (and therefore observations in each statistical type), but also allows us to treat each cluster differently, e.g., by allocating different number of sensors. In addition, since the placement problem is solved independently within each cluster, which is smaller in size, this approach is more scalable than its global counterpart.

4.3. Analysis

In this subsection we analyze and justify the statistical clustering idea. We then discuss how this clustering compares to existing methods.

The following theorem shows that provided we can find mutually independent clusters, there always exists a sensor allocation under which applying the greedy MaxMI

algorithm globally and cluster-by-cluster will generate the same set of selections[8]. The proof can be found in the appendix.

THEOREM 1. *Consider a complete set of locations V, from which we wish to select a set of size K. Assume that we can find C clusters, denoted by V_i, $1 \leq i \leq C$, $V_i \cap V_j = \emptyset$, $\forall i \neq j$ and $\bigcup_i V_i = V$, such that $MI(v, w) = 0$, $\forall v \in C_i, w \in C_j$ and $i \neq j$. Then there exists an allocation $K_i \geq 0$, $1 \leq i \leq C$, $\sum_i K_i = K$, such that $A_K^g = \bigcup_{i=1}^C A_{K_i}^i$, where A_K^g denotes the set (of size K) selected by the greedy MaxMI algorithm applied globally, and $A_{K_i}^i$ denotes the set (of size K_i) selected by the greedy MaxMI algorithm applied to cluster V_i.*

The greedy MaxMI algorithm has been shown to have performance within a constant $(1 - 1/e)$ of the optimal (with respect to the MaxMI objective) by using the submodular property of the objective function [Guestrin et al. 2005; Krause et al. 2007]. The above theorem thus establishes the same performance bound for the clustered greedy MaxMI algorithm under the stated conditions. On the other hand, Theorem 1 assumes the ideal condition that these statistically independent clusters could be found. In practice, this is not likely to be satisfied. If we relax the independence assumption and assume we can find clusters such that the mutual information between locations from different clusters are upper bounded by a threshold ϵ, then results similar to Theorem 1 may be established but will involve more complicated conditions on the relative values of the mutual information between locations within the same cluster, for different clusters. Qualitatively speaking, if ϵ is sufficiently small, then the clustered greedy MaxMI algorithm is a good approximation of its global counterpart.

We next compare the coarse-grained clustering to a well-known clustering method in the statistics and machine learning literature: the K-means algorithm (will be rewritten as C-means in the following as C denotes the number of clusters in our notation). This algorithm partitions N points into C clusters whereby each point belongs to the cluster with the nearest cluster mean or the nearest center of the cluster. More formally, given a set of measurement values x_v for locations $v \in V$, Equations (10) and (11) below find clusters based on the values and the locations' geographical proximity, respectively:

$$\arg \min_{V_i, 1 \leq i \leq C} \sum_{i=1}^{C} \sum_{v \in V_i} ||x_v - \mu_i||^2, \tag{10}$$

$$\arg \min_{V_i, 1 \leq i \leq C} \sum_{i=1}^{C} \sum_{v \in V_i} d(v, Center(V_i)) \ , \tag{11}$$

where μ_i is the mean over all values belonging to cluster V_i, and $Center(V_i)$ is the center point of the area covered by cluster V_i. Note that when each location has a set of values indexed by time, as in our case, we use x_v^t to denote the value observed at time t at location v, and the clustering method given in (10) may be modified to $\arg\min_{V_i, 1 \leq i \leq C} \sum_{t=1}^{T} \sum_i \sum_{v \in V_i} ||x_v^t - \mu_i||^2$, so that the clustering is done over the entire data set. No such changes are needed in (11) as it does not depend on the measurement values. These two methods will be referred to as the *C-means clustering* and *Geo-clustering*, respectively. The computation required in solving Equations (10) and (11) can be quite heavy, especially for large T and $|V|$. Specifically, for $T = 1$ the theoretical

[8]The reason we select MaxMI as the objective is because the greedy MaxMI algorithm has a performance bound which becomes applicable in our case, as shown later.

upper bound of solving (10) is $O(C^{|V|})$ iterations, while Aloise et al. [2009] showed that solving (11) is NP-hard even for instances in the plane.

It turns out that the coarse-grained clustering may be viewed as a very good and highly efficient (simple) approximation of the C-means clustering method. This is formally stated in the following theorem. The proof can be found in the appendix.

THEOREM 2. *Consider a set of values V, $|V| = N$, that we wish to cluster. Let V_i, $1 \leq i \leq C$, denote the optimal clustering result of the C-means problem given in Equation (10), and let $|V_i| = N_i$, $\sum_{i=1}^{C} N_i = N$. Let μ_i denote the mean value of cluster V_i. Without loss of generality, we will assume that $\mu_1 \leq \cdots \leq \mu_C$. Then the N values under the clustering given by $V_1, V_2, \cdots, V_C$ are monotonically nondecreasing, i.e., we have $x_i \leq x_j$, $\forall x_i \in V_i$, $x_j \in V_j$, and $\forall 1 \leq i < j \leq C$.*

The above theorem states that the result of C-means clustering is such that if we order the clusters according to their means, then the values of the cluster members are also monotonically ordered (from one cluster to another; we are not concerned with the ordering within a cluster). In general this does not mean that one can perform C-means by simply ordering the samples, since the ordering of samples at different locations can vary (sometimes drastically) over time. However, in the case of soil moisture data the monotonic ordering is well preserved over time as we have shown in Section 4.1. Thus we can indeed use this ordering to perform clustering, which is exactly what the proposed coarse-grained clustering method does.

The proposed coarse-grained clustering method is statistically consistent with the C-means clustering, as in both cases the clustering will produce a monotonic ordering of the clusters and the member values. Indeed our approach may be viewed as a very simple and effective way of approximating the latter (which is computationally much more demanding) in this case. The advantage of C-means clustering is that it generates the size of each cluster as a result of the optimization (the only input being the number of clusters C and the value set). Under the coarse-grained clustering each cluster is of equal size. On the other hand, C-means clustering cannot determine how many sensors each cluster gets allocated; thus in and by itself C-means clustering does not completely solve the sensor placement problem. Under the coarse-grained clustering, this allocation is determined in the second step by using the the second order statistics (Equation (9)), and this is empirically justified by a similar coarse-grained monotonic ordering of the variances.

In summary, the clustering component of our approach may be viewed as a very good and computationally simple approximation to C-means clustering, while the allocation component of our approach can be used together with C-means clustering to produce an alternative sensor placement solution. This will be numerically evaluated in Section 5. As we will show these two approaches have very similar performance, with the coarse-grained ordering based clustering being much faster computationally. We end this subsection by showing the clustering maps of these two approaches in Figure 5. The similarity between the two is quite obvious. The difference is primarily due to the fact that the cluster sizes are different under these two schemes, as discussed above.

4.4. Discussion

We next discuss the applicability of geography-based clustering (geo-clustering given in (11) is an example) and clustering using physics of soil moisture dynamics.

In some application domains, statistically similar locations are also geographically close, where the statistics are governed by local properties and locations very far away are approximately independent, see e.g., a clustering method proposed in [Krause et al. 2006].

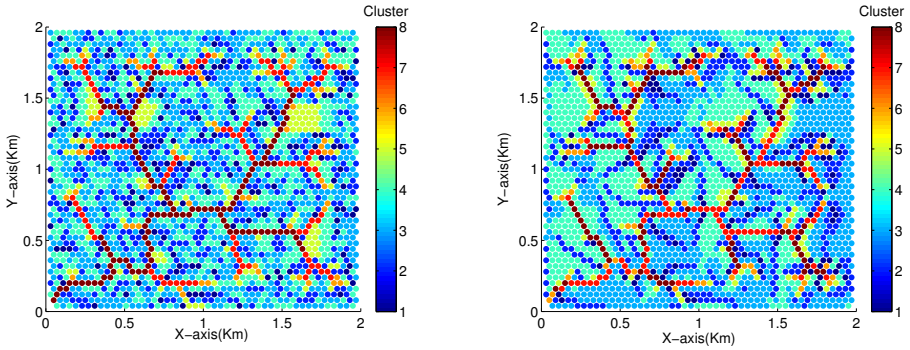


Fig. 5. $C = 8$ color-coded clusters. (a) Coarse-grained clustering. (b) C-means clustering.

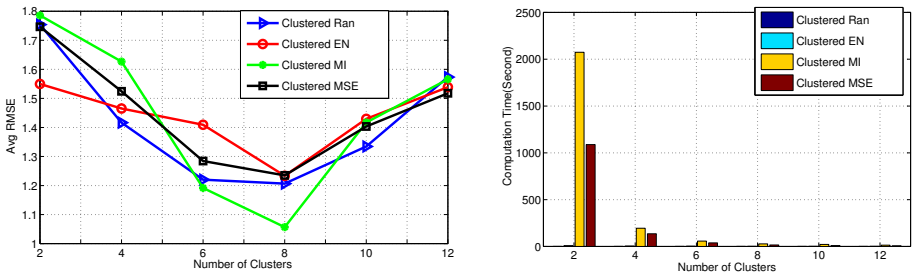
This unfortunately dose not apply to soil moisture sensing. As already mentioned, the dynamic change in soil moisture is dominated by factors including surface energy balance (precipitation, winds, air temperature and humidity, etc.) and soil physics (soil type, topography, and vegetation cover, etc.). These factors can vary significantly in their scale and heterogeneity, and thus locations close by may exhibit drastically different statistical behavior. Discontinuity in landscape can also be a contributing factor. For instance two locations (A, B) along a water way, though far away, may be statistically more similar than that between A and a third location C, which is geographically closer to A but is situated inland and on sandy soil.

There are many such instances in the data set described in Section 3. For example consider the following three locations from the 2D basin area: (30,120), (70,120) and (970,160). The statistical mean and variance at these locations are (148, 168, 148) and (226, 536, 228), respectively. Obviously the first and the third locations are much more similar in these two moments even though the first and the second locations are physically much closer to each other.

In principle, statistically similar locations may be accurately identified through the knowledge of surface energy balance and soil physics. As mentioned, in tRIBS for a single location the time and depth evolution of soil moisture is expressed via a pair of coupled partial differential equations in space and time over these factors. These factors collectively determine the dissipation process of moisture content. It follows that locations with similar surroundings sharing similarity in these factors will exhibit similar dynamics or be highly correlated in their observations. However, these factors are characterized by numerous parameters (the tRIBS simulation requires hundreds of input parameters), resulting in an exponential number of possible combinations, with complex dynamics governing their interactions. Trying to classify locations based on these parameter values is thus infeasible. Therefore, while sound in principle, in practice it is impossible to statistically cluster the locations based on the knowledge of the input to the soil moisture process. The clustering method based on coarse-grained ordering essentially bypasses this difficulty and instead relies on the output of the soil moisture process to infer the statistical similarity among locations.

5. NUMERICAL EXPERIMENTS AND RESULTS

We present in this section a series of numerical experiments performed on the simulated soil moisture data set described in Section 3. We focus on the performance difference between different sensor placement algorithms. We will use Gaussian regression for field estimation given any placement throughout this section. Our performance



(a) (b)

Fig. 6. Performance comparison of coarse-grained clustering scheme with different cluster numbers and placing 300 sensors among 2400 locations. (a) Prediction accuracy. (b) Computation time.

measure is the average root MSE (or AvgRMSE) defined as follows:

$$AvgRMSE = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{T}\sum_{t=1}^{T}\frac{\|\hat{X}_V^t - X_V^t\|_2}{\sqrt{N}}\ , \tag{12}$$

where T is the number of snapshots used to test an algorithm, J is the number of random trials, X_V^t and $\hat{X}_V^t$ denote the vectors of actual and estimated soil moisture readings at all locations in the tth snapshot, respectively, and $\|\ \|_2$ denotes the 2-norm, or the root of the sum of squared errors over all N locations. For placement algorithms that involve random selection of locations, J is set to be 100. For deterministic placement algorithms $J = 1$ as averaging is not needed.

The mean and covariances needed for Gaussian regression and placement algorithms are obtained from a set of T *training snapshots*, using the following sample mean and sample covariance formula: $\bar{\mu}_v = \frac{1}{T}\sum_{t=1}^{T}$ and $\bar{\Sigma}_{u,v} = \frac{1}{T-1}\sum_{t=1}^{T}(x_v^t - \bar{\mu}_v)(x_w^t - \bar{\mu}_w)$, respectively, where x_v^t denotes the observation at location v in the t-th snapshot. The subsequent performance of the algorithms is evaluated using a set of *testing snapshots*. These two sets are mutually exclusive. For most of the experiments presented below we will use the first 1500 snapshots for training and the remaining 708 for testing

We perform two groups of experiments. The first considers only the surface moisture readings and examines the performance of 2D sensor placement schemes. The second considers moisture readings at 3 depths – 2.5 cm (this is also the surface depth considered in 2D), and 19.7 cm and 52.9 cm below surface, respectively – and examine various 3D placement schemes. While the data set contains 9 depths at each location, considering all 9 depths results in very high computational complexity with limited additional benefit or insights comparing to using 3. Our experiments are done for the following criteria/algorithms: MaxEN, MaxMI, and MinMSE, both applied globally and in clusters. As an additional comparison point, we will also present the result of a random placement (referred to as Ran) scheme whereby the locations are selected uniformly randomly out of the set (globally or within a cluster).

5.1. Comparison of Different Clustering Scales

We start by examining the impact the number of clusters C has on the resulting estimation accuracy and computational effort under the proposed coarse-grained ordering based clustering scheme. These are shown in Figures 6(a) and 6(b), respectively.

We see that as the number of clusters increases each cluster becomes smaller, which results in less computation required [9]. This is particularly true for MaxMI and MinMSE. On the other hand, the prediction performance does not behave monotonically as clusters become smaller regardless of which placement methods we use. The accuracy first improves with the reduction in cluster size and then worsens as the cluster size decreases further. This phenomenon in the case of MaxMI may be explained using our analysis in Section 4. When C is very small (say $C = 2$), a large cluster potentially contains statistically very different locations. This can skew the proportional sensor allocation using Equation (9), resulting in poor performance. This is improved as we increase C up to a certain point ($C = 8$). On the other hand, if we continue to increase C, then the correlation between different clusters starts to increase (i.e., it becomes harder to find independent clusters if there is a large number of them). As discussed earlier, higher the correlation, higher the difference between the global and clustered placements. This results in deteriorating performance. Empirically, a good balance seems to be reached at $C = 8$, with 300 locations in each cluster. This will also be the value used for the rest of the experiments.

5.2. Comparison of Different Clustering Schemes

We next compare the performance between three different clustering schemes: the proposed coarse-grained clustering, C-means clustering, and geo-clustering. The number of clusters in each scheme is fixed at $C = 8$. The total number of sensors placed ranges from $K = 50$ to 1000, with increments of 50 in the experiments. For a given K value, under all three schemes the number of sensors allocated to each cluster is done using Equation (9). Under each clustering scheme with a given allocation, we compare the following set of placement algorithms: MaxEN, MaxMI, MinMSE, as well as a purely random scheme Ran.

The performance comparison is shown in Figure 7. Interestingly, the difference between the coarse-grained clustering and the C-means clustering is negligible (compare Figures 7(a) and 7(b)). This similarity has been discussed earlier in Section 4.3. We note that using the same sensor allocation scheme further contributed to the closeness in performance. Under both schemes the three criteria (MaxEN, MaxMI, MinMSE) perform very similarly and do not show obvious performance advantage over the simple random scheme. This is because we allocate fewer (respectively more) sensors to clusters with low (respectively high) mean and variances. Consequently, in the case of a cluster with low variance, all locations behave similarly thus placement according to an objective vs. random placement do not pose significant difference; in the case of a cluster with high variance, more sensors are placed, thus compensating for the difference between the two.

The performance of geo-clustering as shown in Figure 7(c) is obviously inferior to the other two, much as expected. Also in this case deterministic placement perform better than random scheme. This is because under geo-clustering each cluster has similar mean and variance values, resulting in a roughly even distribution of sensors placed in each cluster. Since each cluster contains locations of very different statistical types, random selections of locations clearly becomes inferior to placement that optimizes a performance objective. We also show the computational effort required in dividing all locations into 8 clusters under these three schemes in Figure 7(d). As expected, C-means clustering is the most expensive, followed by geo-clustering. Our proposed clustering method presents significant advantage in this regard.

[9]Thorough this paper, all the numerical experiments are done on an Intel(R) Xeon(R) W3520 2.67GHz processor and 6.00 GHz RAM.

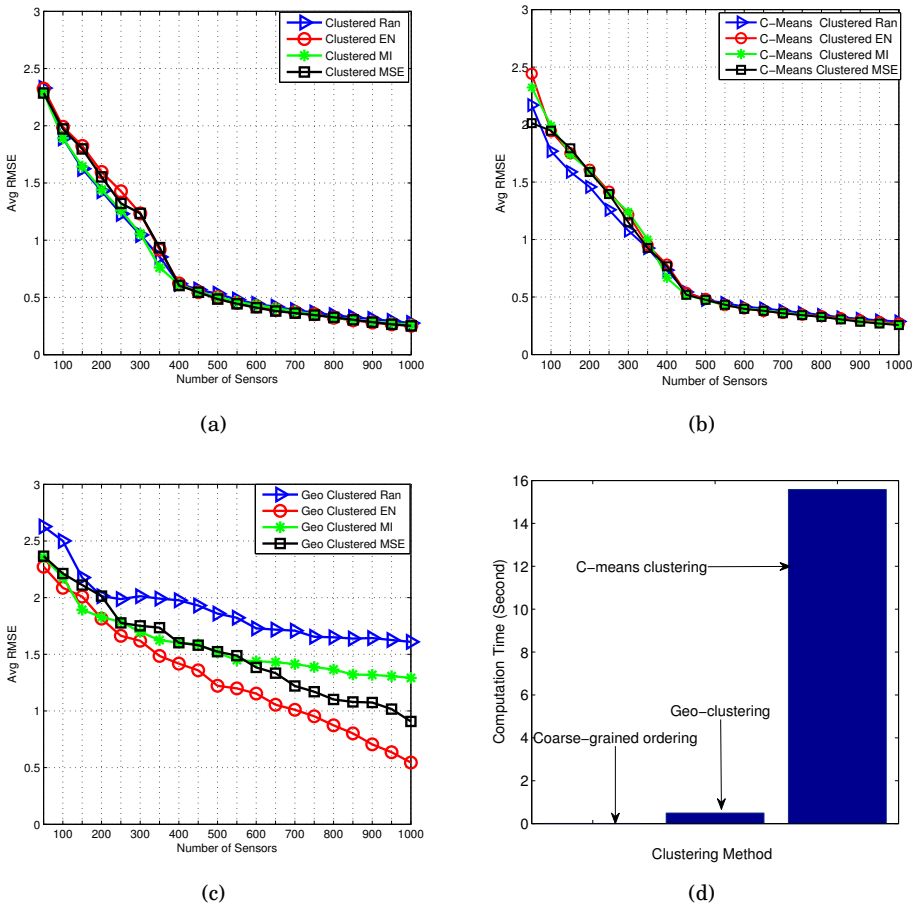


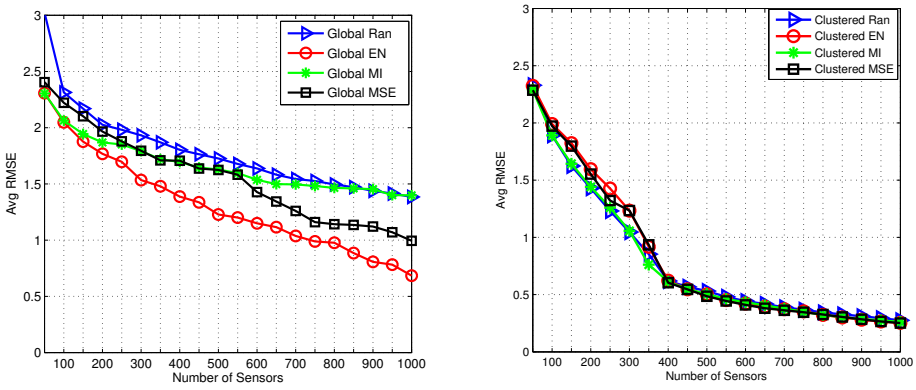
Fig. 7. Performance comparison of different clustering schemes. (a) Coarse-grained clustering. (b) C-means clustering. (c) Geo-clustering. (d) Computation required in generating $C = 8$ clusters.

We conclude that the coarse-grained clustering scheme is competitive against the C-means clustering in terms of prediction performance. We emphasize its two advantages over the latter: (1) It is much simpler in computation. (2) The coarse-grained clustering gives us not only a way to cluster sensors, but also a way to allocate sensors to each cluster. This allocation aspect is absent in the C-means framework.

5.3. Comparison between Clustered and Global Placements

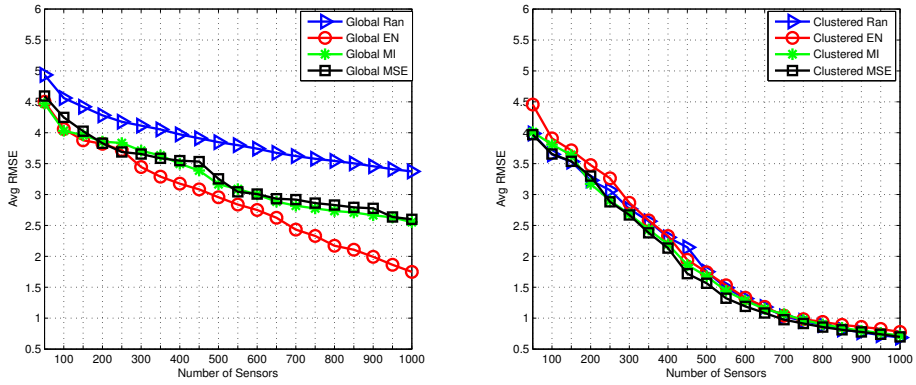
In this section we compare the coarse-grained clustering schemes with their global placement counterparts. Our first set of experiments examines the effect of the number of sensors placed, shown in Figure 8.

Figure 8(a) compares the set of global placement schemes. As expected, the deterministic placements work better than the purely random placement. For the three different criteria, MaxEN performs the best, while MaxMI and MinMSE are very close when $K \leq 500$. This is somewhat contrary to the empirical results reported in Guestrin et al. [2005] that uses a set of indoor building temperature data. MaxMI is generally considered to be better than MaxEN because it can better reduce the uncertainty of



(a) (b)

Fig. 8. Comparison of placement algorithms under different framework. The first 1500 (last 708) are used as training (testing) snapshots. (a) Global Ran/MaxEN/MaxMI/MinMSE. (b) Clustered Ran/MaxEN/MaxMI/MinMSE based on coarse-grained clustering.



(a) (b)

Fig. 9. Comparison of placement algorithms under different framework. The first 800 (last 1408) are used as training (testing) snapshots. (a) Global Ran/MaxEN/MaxMI/MinMSE. (b) Clustered Ran/MaxEN/MaxMI/MinMSE based on coarse-grained ordering clustering.

the unobserved locations. In addition, one would expect MinMSE to perform the best simply because the end performance is measured by RMSE. However, we should note that all three are done using greedy sub-optimal algorithms, and thus the actual performance is a result of the combination between the objective function and the greedy approximation.

By contrast, all clustered placement schemes perform very similarly as shown in Figure 8(b), including the random placement. This suggests that when clustering is adopted, which criterion we use within each cluster becomes less important, as clustering itself already reflects significant feature of the data. Secondly, we see that clustered schemes significantly outperform their global counterparts especially for the clustered random scheme. The performance gain increases as we place more sensors, at least 55% and 70% for clustered MaxEN and clustered Ran, respectively, when $K \geq 700$.

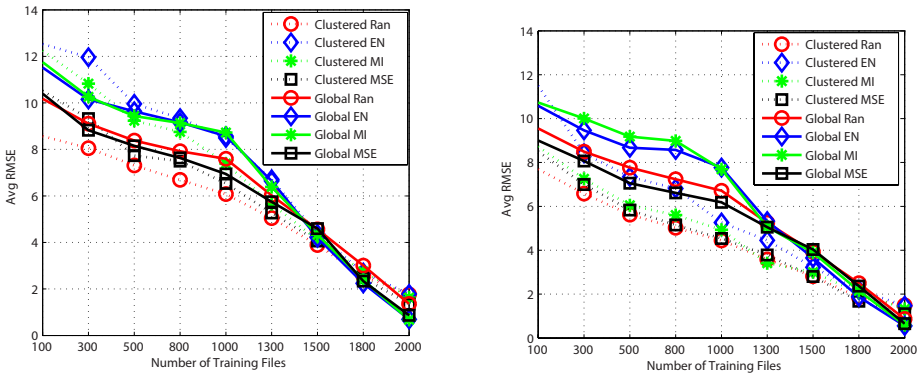


Fig. 10. Impact of training on prediction performance. (a) Placing 100 sensors. (b) Placing 250 sensors.

The same experiments are repeated by using the first 800 snapshots for training and the last 1408 for testing; these are shown in Figure 9. While most of our earlier observations remain the same, we see that as we reduce the amount of training, the performance gain of clustered placement increases, especially when the number of sensors placed is relatively large. For instance, if $K \geq 700$, then the clustering performance gain ranges from 60% (clustered MaxEN) to 75% (clustered Ran).

Our next set of results examines the effect of training more clearly. Figures 10(a) and 10(b) compare the performance of the global placement schemes and clustered schemes by varying the amount of training used, while placing 100 and 250 sensors, respectively. To maintain consistency, in this set of experiments we use the last 208 snapshots for testing regardless of the amount of training used.

The first thing to note is that as the amount of training increases, the performance improves under any scheme, to be expected. Secondly, there is a bigger performance gap between global and clustered placements when the amount of training is less, with clustered placements generally performing better. This is particularly prominent in the case of placing 250 sensors, as shown in Figure 10(b). In all pairs, the clustered placement scheme outperforms the corresponding global scheme, with the exception of MaxEN and MaxMI when using 4% and 14% for training, in the case of placing 100 sensors. When the amount of training exceeds roughly 1500 snapshots in Figure 10(a), and respectively 1800 snapshots in Figure 10(b) (these account for 68% and 82% of the total amount of data), the gain of using clustered placement disappears and global schemes start to show a slight advantage.

These results suggest that clustered placements are more robust to the lack of training, and more robust to temporal variability or non-stationarity present in the data. This is because our clustering is done using the coarse-grained ordering based on relative soil moisture levels (using the training snapshots), a much more stable feature of the data over time than the actual means and covariances. It is therefore much less sensitive to the inaccuracy of the mean and variance statistics generated during training, and more forgiving if the training data does not represent well the testing data. As we use a majority of the data for training, the statistics from the training set become more and more representative of the testing set, thus in this case clustered schemes lose their advantage.

To summarize, we note that sensor placement is entirely an offline process, meaning once sensors are placed their locations cannot be easily adjusted or made adaptable.

Table I. Computation Time (s).

	Clustered	Global
Ran	0.091177	0.0037653
EN	1.395170	10.408
MI	61.686145	103962.00
MSE	21.090523	119943.00

This means that it is highly desirable to have a robust placement scheme, especially when the underlying random process is expected to be non-stationary. In such cases no amount of training may be sufficient, and thus clustered schemes are highly advantageous. In addition, there is significant computational benefit in using clustering. Clustering generally results in 3-4 orders of magnitude reduction in computation time depending on the criterion used, with the saving most significant in the case of MinMSE. Table I shows the computation time (in seconds) incurred by different placement algorithms of choosing K = 250 locations.

5.4. Comparison with Classical Experimental Design Criteria

In the statistics literature, the problem of optimal experimental design has been extensively studied, see e.g., [Atkinson 1988; 1996; Pukelsheim 1987]. The widely used design criteria are A/D/E optimality which maximize the trace/determinant/spectral radius, respectively, of the inverse moment matrix given by $(M^T M)^{-1}$, where $M = (\Sigma_{AU}\Sigma_{UU}^{-1})$, under the linear model $x_A = \Sigma_{AU}\Sigma_{UU}^{-1}x_U + W$. Here $U \subset V$ is a set of locations of interest (and thus used for evaluation), $A \subset V \setminus U$ is the set of selected locations for sensor placement, and W models independent Gaussian measurement noise with constant variance. A detailed description can be found in [Atkinson 1988; 1996]. Since our previous results show that MaxEN provides the best performance in general, in this section we will limit our attention to comparing the global and clustered MaxEN scheme with the A/D/E optimality criteria.

We form the interest set U by randomly selecting 16 locations out 2400. As before all covariances are obtained empirically from the first 1500 training snapshots; and the residual 708 is used as testing snapshots. The $AvgRMSE$ averaged on 50 trials of randomly selected 16 locations is shown by Figure 11. We see that clustered MaxEN is quite competitive compared to the A/D/E criteria, and works better when the number of sensors placed exceeds 300.

5.5. Comparison between Empirical and Exponential Covariance Functions

Throughout our experiments we have used training snapshots to obtain empirical means and covariances between locations. As we have seen the more training we use, the better statistics we collect and this is reflected in the improvement in estimation accuracy under all placement algorithms.

For many spatial phenomena (e.g., temperature, humidity), the covariance of the measurements between two locations is usually considered to be inversely proportional the euclidian distance between them. In geostatistics, a covariance function is often used to capture this relationship; those widely used include exponential,Gaussian, and Matern [Cressie 1993]. Below we use one of these covariance functions, the exponential function, to obtain sensor placement (again using the MaxEN, MaxMI and MinMSE criteria), and compare the performance of this approach with that using empirically obtained covariances. The exponential covariance function (between locations v and w) is given by $\Sigma_{v,w} = \sigma^2 \exp(-d(v,w)/\gamma)$. The unknown parameters (σ^2, γ) are obtained empirically from a training snapshot using maximum likelihood (ML). The comparison results are shown in Figure 12, using 1500 snapshots for training. Similar results were observed with less training.

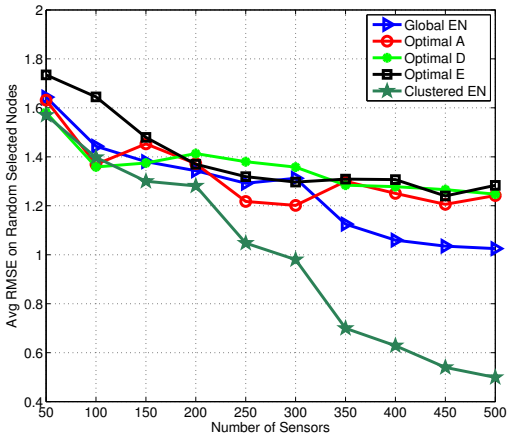
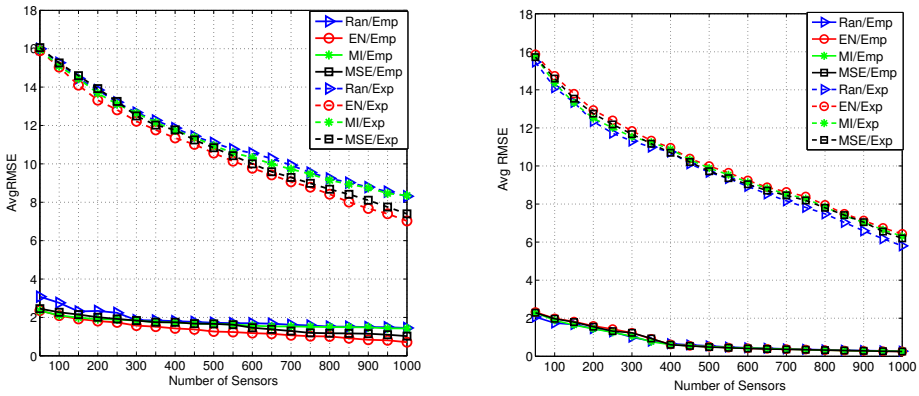


Fig. 11. Comparison between A/D/E optimality criteria and global/clustered MaxEN.



(a) (b)

Fig. 12. Comparison between using empirical and exponential covariances. (a) Global placement schemes. (b) Clustered placement schemes.

We see that using empirical covariances outperforms using this particular covariance function; the performance difference is quite significant and regardless of the type of placement criteria used. As we have observed earlier, due to the heavy influence and variability in external factors like soil types and vegetation cover, the correlation between two locations in general may not be inversely related to their distance. Specifically, instead of diffusing around a neighborhood laterally, water content can dissipate under the surface depending on the soil type, further complicating the relationship [10].

[10]The performance under the exponential covariance function may be improved by sequentially updating the estimates on the unknown parameters (σ^2, γ), as new data become available if the application allows sequential change in the placement, a method studied in [Krause and Guestrin 2007]

5.6. 3D Sensor Placement and Field Estimation

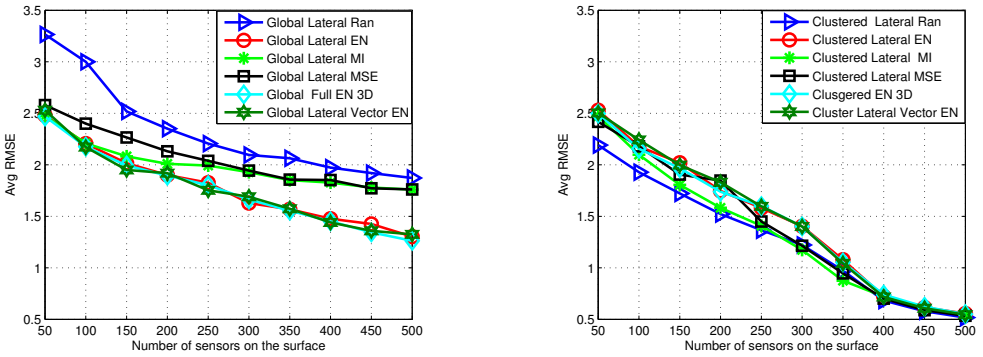
We now turn to the 3D placement problem. From a practical point of view, whenever a location is selected moisture probes are placed at all desired depths (typically between 8 and 10) up to 2 meters deep. Collecting statistics at multiple depths generates a richer description of the moisture process, especially if the soil types/densities vary with depth which is often the case. In addition, placing sensors at all depths also makes sense due to the labor intensive nature of the installation process. This is very much like laying down fiber; the additional cost of placing an extra sensor at an identified location is much lower than creating a new location. This is both a feature and a constraint when solving the placement problem.

Suppose there are L possible depths at each of the N lateral locations, resulting in a total of $L \times N$ possible locations to place sensors (here the term *location* is a point within the 3D domain). The L points at each lateral location are referred to as a *vector* or a *column*. In solving the placement problem, we can either (1) ignore the above constraint, and perform a true 3D placement by selecting LK out of the LN locations and allowing the possibility of having fewer than L sensors at a single lateral location, or (2) take into account the above constraint and select essentially K out of N lateral locations, knowing that at each lateral location we are placing a column of L sensors downward. We will refer to (1) as the *full 3D* placement. Method (2) can further be done in two ways. We can use a single depth to determine the placement of K out of N lateral locations (the MaxEN, MaxMI and MinMSE versions of it) while measuring the estimation error at all depths. Alternatively, we could solve the K out of N placement problem using vectors of size L; this can be done with similar greedy algorithms provided at each step we pick the one vector that maximizes the corresponding objective. We will refer to the former as the *lateral placement* and the latter as the *lateral vector placement*.

Compared to experiments presented in the 2D case, the performance measure in the present case remains the RMSE but averaged over the set of LN points. Due to the limitation of geo-clustering algorithms, below we will only examine the performance of the full 3D, lateral vector, and lateral placement under global and clustered schemes. For the latter the clustering is done by using only the surface level soil moisture data; this is because classifying 3D data is much more complicated and warrants a separate study. We select $L = 3$ out of the 9 layers available in the data for both training and testing. As mentioned earlier, considering all 9 depths results in fairly high computational complexity without much addition insight. In addition, the results shown below are obtained by using the first 1500 snapshots for training and the last 708 snapshots for testing.

Figure 13 compares the performance between global placement schemes and clustered schemes. We see that when the error measure includes all depths, global placement holds no advantage over clustered placement regardless of the number of sensors placed, and the latter presents significant performance gain over the former when K increases. This is a stronger result than that shown in the 2D placement case, where under the same amount of training this advantage emerges when $K > 150$. An additional observation is that using clustering, the difference between objectives MaxEN/MI and MinMSE becomes quite insignificant. Furthermore, lateral vector EN and full 3D EN for each cluster do not have significant performance gap w.r.t. others. This is consistent with an earlier observation: as clustering captures key feature of the data, the objective we use become less important.

We also note that among the global placement schemes, MaxEN outperforms MaxMI and MinMSE, a result consistent with what we observed in the 2D placement case. In addition, MaxEN using the surface data performs very close to the full 3D MaxEN



(a) (b)

Fig. 13. (a) Global placement. (b) Clustered placement.

and lateral vector MaxEN. This is because while the second and the third layers have higher moisture readings, they in general have much lower variances. Therefore the difference between full 3D and 2D surface placements using MaxEN becomes negligible.

6. DISCUSSION AND PRACTICAL IMPLICATIONS

The classes of sensor placement algorithms studied in this paper heavily rely on the availability of the statistics of the field in which we wish to deploy sensors. While this is a very commonly used assumption, it needs to be more carefully justified. Such an assumption sometimes leads to a chicken-and-egg question: don't we need finely deployed sensors to collect such statistics to begin with? In the context of soil moisture sensing, such fine-grained statistics based on in-situ sensing is currently unavailable as we have mentioned earlier. However, advanced soil moisture and landscape simulators exist, and as we have shown one can use simulated data to test the performance of a placement algorithm prior to deployment.

Specifically, to use the algorithms studied here in practice, we will first need to decide on an area or sensing field. We then collect information about the area to provide as inputs to soil moisture simulators; such information includes geographical location, landscape, soil types, vegetation covers, historical data on rainfall, among many other factors. Using these inputs, the simulator generates spatial and temporal soil moisture data. This is then used to determine a subset of locations to place sensors as we have described in previous sections. In some cases we may be able to use collected data from a few locations to calibrate the simulator before larger scale sensor deployment. As we have observed, since no real data (of a reasonable scale) is available, it is particularly important to have a placement scheme that is robust to variations in training data or the lack of training data. In this sense the proposed class of clustered placement schemes offers a very good solution in terms of accuracy, robustness and scalability. It's worth noting that the fundamental notion behind the clustering scheme is much more general and broadly applicable than soil moisture sensing – it points to the idea that if we base placement decisions on statistical behavior, then grouping statistically similar locations together results in higher efficiency on a macroscopic level, while at the same time allowing us to treat each location within the same cluster similarly (recall that we order the locations randomly within each cluster). This in turn results in more robustness at the microscopic level.

7. CONCLUSIONS

In this paper we studied the problem of optimal sensor placement and field estimation for the application of soil moisture sensing. We showed that soil moisture data admits a coarse-grained monotonic ordering of locations in terms of their soil moisture content, a stable feature over time, that can be exploited in designing good sensor placement schemes. Using this feature we proposed a class of clustered sensor placement schemes. Extensive numerical results were presented and different sensor placement schemes were evaluated. We conclude that the coarse-grained ordering of locations is a far more stable feature inherent in the soil moisture data, that leads to much more scalable and robust placement algorithms. In addition, these algorithms in general outperform those solely relying on the Gaussian assumption.

APPENDIX

A. PROOF OF THEOREM 1

We prove this theorem by induction on K.

Induction basis: Consider $K = 1$; then A_1^g consists of a single location/element denoted as v^*. Using the greedy MaxMI algorithm it is obtained as follows:

$$\begin{aligned} v^* &= \arg\max_{v\in V} MI(v, V\setminus v) \\ &= \arg\max_{v\in V}[H(V\setminus v) - H(V\setminus v|v)] \qquad (13) \\ &= \arg\max_{v\in V_i; 1\le i\le C}[H(V\setminus v) - H(V\setminus v|v)]\,, \qquad (14) \end{aligned}$$

where the third equality is due to the fact that v^* must be from one of the C clusters. Using the fact that locations from different clusters are mutually independent and the additivity of entropy of independent subsets, the two terms in (14) can be written as follows, respectively, for $v \in V_i$:

$$\begin{aligned} H(V\setminus v) &= H[\cup_{j\neq i}^{C} V_j \cup (V_i\setminus v)] \\ &= \sum_{i\neq j}^{C} H(V_j) + H(V_i\setminus v); \\ H(V\setminus v|v) &= H[(\cup_{j\neq i}^{C} V_j \cup (V_i\setminus v))|v] \\ &= \sum_{j\neq i}^{C} H(V_j) + H(V_i\setminus v|v)\,. \end{aligned}$$

Therefore, continuing from (14), we have

$$\begin{aligned} v^* &= \arg\max_{v\in V_i; 1\le i\le C} H(V_i\setminus v) - H(V_i\setminus v|v) \qquad (15) \\ &= \arg\max_{v\in V_i; 1\le i\le C} MI(v, V_i\setminus v) \qquad (16) \end{aligned}$$

That is, the greedy global MaxMI selection v^* may be viewed as the outcome of selecting the largest among all greedy MaxMI selections within individual clusters. Without loss of generality, suppose $v^* \in V_j$, then the above result demonstrates that there exists allocation K_i, where $K_i = 0$ for $i \neq j$ and $K_j = 1$, such that $A_1^g = \{v^*\} = A_1^j = \cup_{i\neq j}^{C} A_{K_i}^i \cup A_{K_j}^j$. The induction basis is thus established.

Induction basis: Now assume the result is true for greedily selecting K locations. That is, $\exists K_i, 1 \le i \le C, \sum_{i=1}^{C} K_i = K$, such that $A_K^g = \cup_{i=1}^{C} A_{K_i}^i$. We next show that the same holds for selecting $K+1$ locations.

Since we are using the greedy MaxMI algorithm, we have $A^g_{K+1} = A^g_K + \{v^*\}$ for some location $v^* \in V \setminus A^g_K$. The selection of v^* is given by

$$\begin{aligned} v^* &= \arg\max_{v \in V \setminus A^g_K} MI[A^g_K \cup v, V \setminus (A^g_K \cup v)] \\ &= \arg\max_{v \in V \setminus A^g_K} MI[\cup_{i=1}^C A^i_{K_i} \cup v, V \setminus (\cup_{i=1}^C A^i_{K_i} \cup v)] \quad (17) \\ &= \arg\max_{v \in V \setminus A^g_K} H[V \setminus (\cup_{i=1}^C A^i_{K_i} \cup v)] - H[V \setminus (\cup_{i=1}^C A^i_{K_i} \cup v) | (\cup_{i=1}^C A^i_{K_i} \cup v)] \\ &= \arg\max_{v \in V^i \setminus A^i_{K_i}; 1 \le i \le C} H[B] - H[B | (\cup_{j=1}^C A^j_{K_j} \cup v)] \quad (18) \end{aligned}$$

where we have use the notation $B = V \setminus (\cup_{i=1}^C A^i_{K_i} \cup v)$, and the second equality (17) is due to the induction hypothesis. As before, using the fact that locations from different clusters are mutually independent and the additivity of entropy of independent sets, we have that for $v \in V^i \setminus A^i_{K_i}$:

$$\begin{aligned} H[B] &= \sum_{j \neq i} H(V_j \setminus A^j_{K_j}) + H(V_i \setminus (A^i_{K_i} \cup v)) \\ H[B | (\cup_{j=1}^C A^j_{K_j} \cup v)] &= \sum_{j \neq i} H(V_j \setminus A^j_{K_j} | A^j_{K_j}) + H(V_i \setminus (A^i_{K_i} \cup v) | (A^i_{K_i} \cup v)) \end{aligned}$$

Continuing from (18), we have

$$v^* = \arg\max_{v \in V_i \setminus A^i_{K_i}; 1 \le i \le C} \left\{ \sum_{j \neq i}^{C} MI(A^j_{K_j}, V_j \setminus A^j_{K_j}) + MI(A^i_{K_i} \cup v, V_i \setminus (A^i_{K_i} \cup v)) \right\} \quad (19)$$

Clearly, the $(K+1)$th element v^* is selected from one of the clusters. For any cluster V_i, the first term on the RHS of (19) is independent of the selection of the location $v \in V_i$. The selected location within cluster V_i has to be such that it maximizes the second term, which is the exactly the same criterion used by the greedy MaxMI algorithm applied to cluster V_i. v^* is then determined by selecting the best across all clusters. Without loss of generality, suppose $v^* \in V_j$. Then the above result demonstrates that there exists an allocation $K_i^{'}$, where $K_i^{'} = K_i$ for $i \neq j$, and $K_j^{'} = K_j + 1$, such that $\sum_i K_i^{'} = K + 1$, and

$$A^g_{K+1} = A^g_K \cup v^* = \cup_{i \neq j} A^i_{K_i} \cup (A^j_{K_j} \cup v^*) = \cup_{i \neq j} A^i_{K_i} \cup A^j_{K_j+1} = \cup_i A^i_{K_i'}.$$

This completes the induction step and thus the proof.

B. PROOF OF THEOREM 2

We prove this theorem by contradiction. Assume the monotonicity does not hold, which means there must exist two clusters V_i, V_j, with $\mu_i \le \mu_j$, each having an element x_{il} and x_{jh}, denoting the l-th element of V_i and the h-th element of V_j, respectively, such that $x_{il} > x_{jh}$.

The summation in the C-means formulation (10) under the optimal clustering $V_i, V_2, \cdots, V_C$ is given by:

$$\sum_{c=1}^{C}\sum_{n=1}^{N_c}(x_{cn}-\mu_c)^2 = \sum_{c\neq i,j}^{C}\sum_{n=1}^{N_c}(x_{cn}-\mu_c)^2 + \sum_{n=1}^{N_i}(x_{in}-\mu_i)^2 + \sum_{n=1}^{N_j}(x_{jn}-\mu_j)^2 \ . \quad (20)$$

Consider now a new clustering method, where we switch the membership of x_{il} and x_{jh} while keeping everything else the same. Denote the new clusters by $V_i^{'}$, the mean $\mu_i^{'}$, $1 \leq i \leq C$, and the n-th element of $V_i^{'}$ by $x_{in}^{'}$. The summation in (10) under this new clustering scheme is given by:

$$\sum_{c=1}^{C}\sum_{n=1}^{N_c}(x_{cn}^{'}-\mu_c^{'})^2 = \sum_{c\neq i,j}^{C}\sum_{n=1}^{N_c}(x_{cn}-\mu_c)^2 + \sum_{n=1}^{N_i}(x_{in}^{'}-\mu_i^{'})^2 + \sum_{n=1}^{N_j}(x_{jn}^{'}-\mu_j^{'})^2 \ , \quad (21)$$

where

$$\begin{cases} x_{in}^{'} = x_{jh} & n = l \\ x_{in}^{'} = x_{in} & n \neq l \end{cases} \qquad \begin{cases} x_{jn}^{'} = x_{il} & n = h \\ x_{jn}^{'} = x_{jn} & n \neq h \end{cases}$$

Taking the difference between these two sums we get:

$$\begin{aligned} & (20) - (21) \\ = \ & (x_{il})^2 - (x_{jh})^2 + 2(\mu_i^{'} - \mu_i)\sum_{n\neq l}^{N_i} x_{in} + 2x_{jh}u_i^{'} - 2x_{il}\mu_i + N_i[(\mu_i)^2 - (\mu_i^{'})^2] \\ + \ & (x_{jh})^2 - (x_{il})^2 + 2(\mu_j^{'} - \mu_j)\sum_{n\neq h}^{N_j} x_{jn} + 2x_{il}\mu_j^{'} - 2x_{jh}\mu_j + N_j[(\mu_j)^2 - (\mu_j^{'})^2] \ . \end{aligned} \quad (22)$$

Noting

$$\mu_i = \frac{x_{i1}+\cdots x_{il}+\cdots+x_{iN_i}}{N_i} \ , \qquad \mu_j = \frac{x_{j1}+\cdots x_{jh}+\cdots+x_{jN_j}}{N_j} \ ,$$
$$\mu_i^{'} = \frac{x_{i1}+\cdots x_{jh}+\cdots+x_{iN_i}}{N_i} \ , \qquad \mu_j^{'} = \frac{x_{j1}+\cdots x_{il}+\cdots+x_{jN_j}}{N_j} \ ,$$

(22) can be further simplified to

$$\begin{aligned} (22) \ &= \ (x_{il}-x_{jh})[x_{il}+x_{jh}-\mu_i-\mu_i^{'}] + (x_{jh}-x_{il})[x_{jh}+x_{il}-\mu_j-\mu_j^{'}] \\ &= \ (x_{jh}-x_{il})[(\mu_i+\mu_i^{'}) - (\mu_j+\mu_j^{'})] \ . \end{aligned} \quad (23)$$

Since $\mu_i \leq \mu_j$ and $x_{il} > x_{jh}$, we must have $\mu_i^{'} < \mu_i \leq \mu_j < \mu_j^{'}$. We thus conclude (22) = (20) − (21) > 0. This means that the new clustering method results in a smaller summation in (10), which contracts the fact that the original clustering method is optimal. Therefore the monotonicity holds, completing the proof.

ACKNOWLEDGMENTS

This work was carried out at the University of Michigan as part of the following two projects "Soil Moisture Smart Sensor Web Using Data Assimilation and Optimal Control" (NASA grant NNX06AD47G) and "Ground Network Design And Dynamic Operation For Near Real-Time Validation of Space-Borne Soil Moisture Measurements" (NASA grant NNX09AE91G), both through the Earth Science Technology Office, Advanced Information Systems Technologies program (AIST). The authors would like to thank other members

of the project team: Profs. M. Moghaddam and D. Teneketzis, Y. Goykhman, A. Nayyar and D. Shuman for their helpful comments and suggestions. The authors specially thank team members Prof. D. Entekhabi and A. Flores for giving access to the data upon which the analysis in this paper is based. The authors would also like to thank the anonymous reviewers and the associate editor for comments and suggestions that greatly helped improve the quality of the manuscript. X. Wu also thanks UESTC and CSC in providing her with the support and opportunity to study at the University of Michigan as a visiting student.

REFERENCES

AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI, E. 2002. A survey on sensor networks. *IEEE Communications Magazine 40,* 8, 102–114.

ALOISE, D., DESHPANDE, A., HANSEN, P., AND POPAT, P. 2009. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning 75,* 2, 245–248.

ATKINSON, A. C. 1988. Recent developments in the methods of optimum and related experimental designs. *International Statistical Review/Revue Internationale de Statistique 56,* 2, 99–115.

ATKINSON, A. C. 1996. The usefulness of optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological) 58,* 1, 59–76.

BELL, K., BLANCHARD, B., SCHMUGGE, T., AND WITCZAK, M. 1980. Analysis of surface moisture variations within large-field sites. *Water Resource Research 16,* 4, 796–810.

BIAN, F., KEMPE, D., AND GOVINDAN, R. 2006. Utility-based sensor selection. In *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, New York, NY, USA, 11–18.

BYERS, J. AND NASSER, G. 2000. Utility-based decision-making in wireless sensor networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. IEEE, Piscataway, NJ, USA, 143–144.

CASELTON, W. AND ZIDEK, J. 1984. Optimal monitoring network designs. *Statistics and Probability Letters 2,* 4, 223–227.

CASELTON, W. F. AND HUSAIN, T. 1980. Hydrologic networks: Information transmission. *Journal of Water Resources Planning and Management 106,* 2, 503–520.

CHOI, H. L. 2009. Adaptive sampling and forecasting with mobile sensor networks. Ph.D. thesis, Massachusetts Institute Of Technology, Bonn, Germany.

COSH, M., THOMAS, J., RAJAT, B., AND PRUEGER, J. 2004. Watershed scale temporal stability of soil moisture and its role in validation satellite estimates. *Remote Sensing Environ. 92,* 4, 427–435.

CRESSIE, N. A. C. 1993. *Statistics for Spatial Data*. Wiley Interscience, New York.

DAS, A. AND KEMPE, D. 2008a. Algorithms for subset selection in linear regression. In *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 45–54.

DAS, A. AND KEMPE, D. 2008b. Sensor selection for minimizing worst-case prediction error. In *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, Washington, DC, USA, 97–108.

D.VINOD, H. 1969. Integer programming and the theory of grouping. *Journal of the American Statistical Association 64,* 326, 506–619.

EFRON, B. AND TIBSHIRANI, R. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.

ERTIN, E., FISHER, J. W., AND POTTER, L. C. 2003. Maximum mutual information principle for dynamic sensor query problems. In *Second International Workshop on Information Processing in Sensor Networks*. ACM, New York, NY, USA, 405–416.

FAMIGLIETTI, J., DEVEREAUX, J., LAYMON, C., TSEGAYE, T., HOUSER, P., JACKSON, T., GRAHAM, S., RODELL, M., AND OEVELEN, P. V. 1999. Ground-based investigation of soil moisture variability within remote sensing footprints during the southern great plains 1997 (sgp97) hydrology experiment. *Water Resource Research 35,* 6, 1839–1851.

FLORES, A., IVANOV, V., ENTEKHABI, D., AND BRAS, R. 2009. Impact of hillslope-scale organization of topography, soil moisture, soil temperature and vegetation on modeling surfae microwave radiation emission. *IEEE Transactions on Geoscience and Remote Sensing 47,* 8, 2557–2571.

FRANCIS, C., THORNES, J., ROMERO-DIAZ, A., LOPEZ-BERMUDEZ, F., AND FISHER, G. 1986. Topographic control of soil moisture, vegetation cover and land degradation in a moisture stressed mediterranean environment. *Catena 13,* 2, 211–225.

GONZÁLEZ-BANOS, H. 2001. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry*. ACM, New York, NY, USA, 232–240.

GUESTRIN, C., KRAUSE, A., AND SINGH, A. P. 2005. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*. ACM, New York, NY, USA, 265–272.

HAWLEY, M., JACKSON, T. J., AND MCCUEN, R. 1983. Surface soil moisture variation on small agricultural watersheds. *Journal of Hydrology 62,* 4, 179–200.

HILLS, R. AND REYNOLDS, S. 1969. Illustrations of soil moisture variability in selected areas and plots of different sizes. *Journal of Hydrology 8,* 1, 27–47.

HOCHBAUM, D. AND MAAS, W. 1985. Approximation schemes for covering and packing problems in image processing and vlsi. *J. ACM 32,* 1, 130–136.

JAIN, A. K., MURTY, M. N., AND FLYNN, P. J. 1999. Data clustering: a review. *ACM Computing Surveys 31,* 9, 264–323.

KEMPPAINEN, A., MAKELA, T., HAVERINEN, J., AND RONING, J. 2008. An experimental environment for optimal spatial sampling in a multi-robot system. In *The 10th International Conference on Intelligent Autonomous Systems (IAS-10)*. ACM, Baden Baden, Germany, 265–272.

KO, C., LEE, J., AND QUEYRANNE, M. 1995. An exact algorithm for maximum entropy sampling. *Operations Research 43,* 4, 684–691.

KRAUSE, A. AND GUESTRIN, C. 2007. Nonmyopic active learning of gaussian processes – an exploration–exploitation approach. In *Proceedings of the 24th International Conference on Machine Learning*. ACM, New York, NY, USA, 449–456.

KRAUSE, A., GUESTRIN, C., GUPTA, A., AND KLEINBERG, J. 2006. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, New York, NY, USA, 2–10.

KRAUSE, A., SINGH, A., AND GUESTRIN, C. 2007. Near-optimal sensor placements in gaussian processes: Theory, effecient algorithms and empirical studies. Technical report, Machine Learning Department, Carnegie Mellon University.

LINDLEY, D. V. 1956. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics 27,* 4, 986–1005.

MACQUEEN, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley, CA, USA, 281–297.

MARTINEZ-FERNANDEZ, J. AND CEBALLOS, A. 2005. Mean soil moisture estimation using temporal stability analysis. *Journal of Hydrology 3,* 12, 28–38.

MOGHADDAM, M., ENTEKHABI, D., GOYKHMAN, Y., LI, K., LIU, M., MAHAJAN, A., NAYYAR, A., SHUMAN, D., AND TENEKETZIS, D. 2010. A wireless soil moisture smart sensor web using physics-based optimal control:concept and initial demonstrations. *International Journal of Geographical Information Science 3,* 4, 522–535.

NASA 2006. Nasa strategic plan. http://www.nasa.gov/.

PUKELSHEIM, F. 1987. Information increasing orderings in experimental design theory. *International Statistical Review/Revue Internationale de Statistique 52,* 2, 203–219.

SHUMAN, D., NAYYAR, A., MAHAJAN, A., GOYKHMAN, Y., LI, K., LIU, M., TENEKETZIS, D., MOGHADDAM, M., AND ENTEKHABI, D. 2010. Measurement scheduling for soil moisture sensing:from physical models to optimal control. In *Proceedings of the IEEE Special Issue on Sensor Networks and Applications*. ACM, New York, NY, USA, 918–1933.

SMAP 2008. The soil moisture active and passive mission (smap). http://smap.jpl.nasa.gov/.

VIVONI, E., TELES, V., IVANOV, V., BRAS, R., AND ENTEKHABI, D. 2005. Embedding landscape processes into triangulated terrain models. *International Journal of Geographical Information Science 19,* 4, 429–457.

WESTERN, A., GRAYSON, R. B., AND BLOSCHL, G. 2002. Scaling of soil moisture: A hydrologic perspective. *Annual Review of Earth and Planetary Sciences 30,* 1, 149–180.

WILSON, D. J., WESTERN, A. W., GRAYSON, R. B., BERG, A. A., LEAR, M. S., RODELL, M., FAMIGLIETTI, J. S., WOODS, R. A., AND MCMAHON, T. A. 2003. Spatial distribution of soil moisture over 6 and 30 cm depth, mahurangi river catchment, new zealand. *Journal of Hydrology 276,* 1, 254–274.

YANG, Y. AND BLUM, R. S. 2008. Sensor placement in gaussian random field via discrete simulation optimization. *IEEE Signal Processing Letters 15,* 3, 729–732.

Received June 2010; revised December 2010; accepted XXXXXX