Title
Call and Response: Experiments in Sampling the Environment

Authors
Batalin, Maxim
Rahimi, Mohammed
Yu, Yan
et al.

Publication Date
2004-05-05

Call and Response: Experiments in Sampling the Environment

Maxim A. Batalin[1,2], Mohammad Rahimi[1,3], Yan Yu[1,3], Duo Liu[1,4], Aman Kansal[1,4], Gaurav S. Sukhatme[1,2], William J. Kaiser[1,4], Mark Hansen[1,5], Gregory J. Pottie[1,4], Mani Srivastava[1,4],Deborah Estrin[1,3]

[1]Center for Embedded Networked Sensing, UCLA, Los Angeles, CA 90095
[2]Department of Computer Science, USC, Los Angeles, CA 90089
[3]Department of Computer Science, UCLA, Los Angeles, CA 90095
[4]Department of Electrical Engineering, UCLA, Los Angeles, CA 90095
[5]Department of Statistics, UCLA, Los Angeles, CA 90095

{maxim@sagan.usc.edu, mhr@cens.ucla.edu, yanyu@cs.ucla.edu, duoliu@ee.ucla.edu, gaurav@robotics.usc.edu, kaiser@ee.ucla.edu, cocteau@stat.ucla.edu, pottie@ee.ucla.edu, destrin@cs.ucla.edu}

***Abstract*— Monitoring of environmental phenomena with embedded networked sensing confronts the challenges of both unpredictable variability in the spatial distribution of phenomena, coupled with demands for a high spatial sampling rate in three dimensions. For example, low distortion mapping of critical solar radiation properties in forest environments may require two-dimensional spatial sampling rates of greater than 10 $samples/m^2$ over transects exceeding 1000 m^2. Clearly, adequate sampling coverage of such a transect requires an impractically large number of sensing nodes. This paper describes a new approach where the deployment of a combination of autonomous-articulated and static sensor nodes enables sufficient spatiotemporal sampling density over large transects to meet a general set of environmental mapping demands.**

To achieve this we have developed an embedded networked sensor architecture that merges sensing and articulation with adaptive algorithms that are responsive to both variability in environmental phenomena discovered by the mobile sensors and to discrete events discovered by static sensors. We begin by describing the class of important driving applications, the statistical foundations for this new approach, and task allocation. We then describe our experimental implementation of adaptive, event aware, exploration algorithms, which exploit our wireless, articulated sensors operating with deterministic motion over large areas. Results of experimental measurements and the relationship among sampling methods, event arrival rate, and sampling performance are presented.

I. Introduction

A broad class of environmental monitoring objectives in fundamental science, environmental resource management, and public health protection demand distributed sensing capabilities [1]. For example, the potential global impact of climate change creates a requirement for understanding the interaction between the forest canopy and the atmosphere. Understanding critical phenomena, for example the nature of carbon flux from the atmosphere to forest biomass, requires direct experimental characterization of spatiotemporally distributed phenomena [2]. This includes measurement of solar radiation (driving fundamental photosynthesis and growth), atmospheric water vapor, temperature, and chemical composition. Distributed measurements are required since these phenomena are sensitive to (and in turn influence) the heterogeneous structure of the natural environment.

A. High Fidelity Environmental Field Monitoring

Many environmental monitoring applications share the challenge for achieving high fidelity characterization capability for environmental field variables. This is conventionally achieved with high spatiotemporal sampling rate. For example, solar radiation and atmospheric properties must be mapped in natural environments over a spatial extent comparable to those of the forest structure (where a measurement transect height may exceed 50m and the width required to adequately span the heterogeneous structure may exceed 100m) . At the same time, some phenomena display a characteristic spatial variability on the scale of centimeters, reflecting the fine scale of natural structure and important phenomena. Thus, in considering high fidelity sampling for these phenomena over a two-dimensional plane with the required spatial extent of over 1000 m^2 and resolution of greater than 10 $sample/m^2$ requires an impractically large number of sensing elements with 10,000 measurement points.

Measurement distortion may also result from an improper spatial-sampling distribution (specifically due to mismatch between the spatial structure of the phenomena and sensor node placement). This introduces yet further challenges for environmental monitoring by limiting the capability for distributed sensor node deployment planning. Specifically, the inherent unpredictability and variability of environmental structure and phenomena precludes the possibility of achieving adequate spatial sampling density by advance planning. The conventional

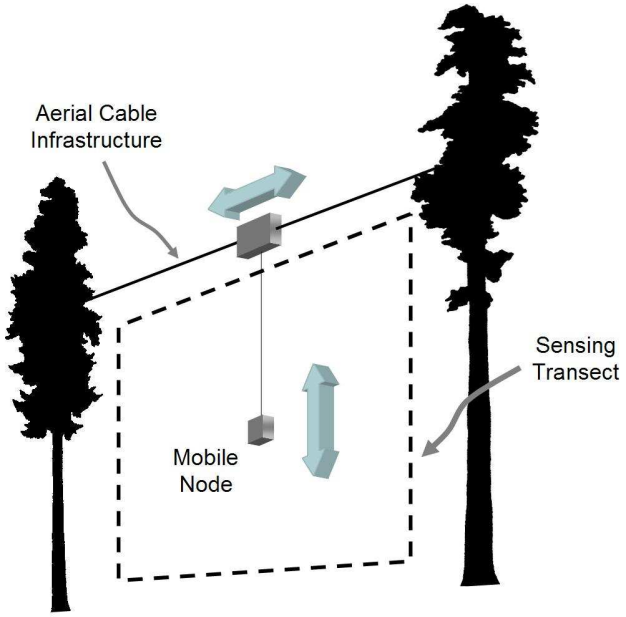


Fig. 1. The NIMS system exploits suspended aerial infrastructure to enable sustainable and precise transport of mobile nodes within complex three-dimensional environments. Phenomena are mapped within a spatially extensive transect where sensing nodes have precise command over horizontal and vertical motion.

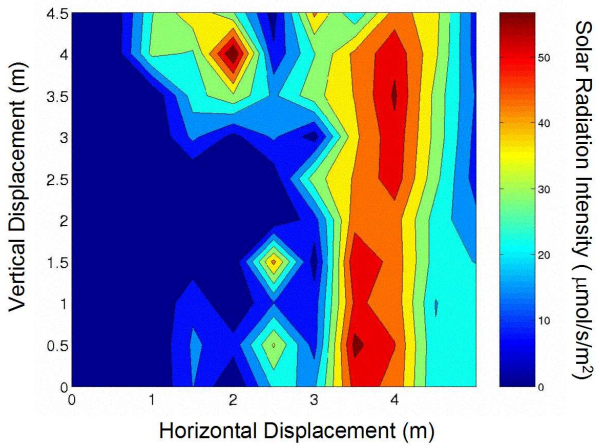


Fig. 2. Map of solar radiation intensity obtained in a forest ecosystem by a NIMS system transporting a light intensity sensor within the canopy. Solar radiation intensity is indicated in contours as it varies spatially according to horizontal and vertical sensor displacement.

solution for reducing this source of distortion has been to increase spatiotemporal sampling rate. However, this results in excessive disturbance to the sensed environment. Clearly, fixed sensor distributions, alone, are not adequate for many important environmental monitoring phenomena.

Since it is the presence of dynamic physical structures that lead to unpredictable and variable sensor coverage requirements, sensor networked systems must exploit wide-ranging and high-spatial-resolution sensor node *mobility* in order to ensure adequate coverage.

B. Coordinated Fixed and Mobile Sensing Nodes for High Fidelity Coverage

This paper describes a new Networked Infomechanical Systems (NIMS) architecture that combines *both fixed and mobile* sensor nodes to achieve a spatiotemporal environment coverage that is dramatically advanced over that of either system alone. Mobility allows the networked sensor system to always seek the most efficient spatiotemporal sampling distribution to achieve a *specified* accuracy of environmental variable reconstruction. Further, mobility also permits the NIMS system to respond to initially unpredictable and variable environmental evolution.

While it is shown here that this architecture enables mobile sensors that adapt to variable environments, their coverage at any instant is restricted to their effective area of regard. Thus, temporal measurement distortion will appear in the presence of rapidly changing phenomena.

The NIMS architecture introduces coordination between fixed and mobile devices. Sparsely distributed fixed sensor nodes, each remaining vigilant over a localized area of regard, provide a distributed event detection service for the combined network. We experimentally demonstrate that proper task allocation of mobile devices permits both high spatial sampling rate and low event response time.

The NIMS architecture coordinating mobile and fixed devices provides the ability to accommodate high fidelity sensing with limited resource constraints. For example, typical environmental monitoring characterization tasks require costly sensing modalities where the cost is measured in terms of energy, mass, volume, monetary expense, or other value). Examples in natural environmental monitoring are tasks that require gas phase analysis or multi-spectral imaging. Thus, while the demand for dense spatial coverage is high, the resource cost associated with supporting a vast number of individual sensor elements for adequate spatiotemporal coverage may be excessively high and the disturbance to the environment due to their introduction may be prohibitive. However, it is demonstrated here that the combination of low spatial-density fixed sensors (providing the required assets for mobile node task allocation) and highly capable mobile devices results in both adequate spatiotemporal coverage as well reduced overall resource cost.

The NIMS architecture for mobile and adaptive sampling must include the attributes of 1) wide range mobility within three-dimensional volumes, 2) precise and high resolution position and orientation control, 3) long term autonomous, sustainable operation, and 4) mobile operation without disturbance to the environment. The combination of these requirements are met by NIMS

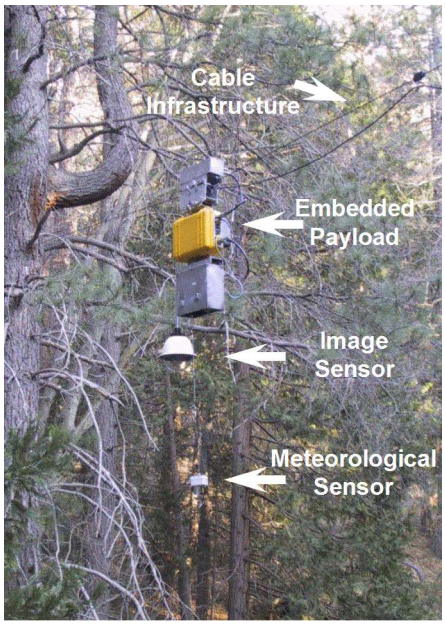


Fig. 3. A NIMS system deployed at the James San Jacinto Mountain Reserve [4]. This image shows the NIMS cable infrastructure, horizontal transport node (carrying an embedded computing platform, image sensor, and vertical transport control, and vertically mobile meteorological sensor node.

infrastructure-supported mobility, as shown in Figure 1. Here, infrastructure is adapted to the environment and uniquely enables each of the attributes above.

C. Motivating Application

An example of a fundamental phenomenon that controls processes within natural ecosystems is the flux of solar radiation. Solar radiation is spatially filtered by the complex ecosystem structure and ultimately controls photosynthesis and growth [3]. The characterization of solar radiation spatiotemporal patterns is of primary interest to understanding growth, evolution, and global change trends [2].

NIMS provides the first method for extensive spatiotemporal mapping of phenomena, including solar radiation in the ecosystem. Figure 2 displays a map of solar illumination intensity in the photosynthetically active spectral region (PAR). This two dimensional PAR map was acquired by a NIMS system operating in a mixed conifer forest within the James San Jacinto Mountain Reserve [4]. Note the characteristic scale of variability for solar radiation is less than 1m with a need to characterize a spatial extent that equals the dimensions of the forest canopy itself.

Figure 3 shows a NIMS system deployed in the forest reserve for continuous operation. This system includes supporting cable infrastructure, a horizontally mobile embedded computing platform payload, image sensing, and a vertically mobile meteorological sensor system carrying water vapor, temperature, and PAR sensing capability. Wireless networking is incorporated to link fixed nodes (distributed on the surface or suspended) with the vertically and horizontally mobile elements. The NIMS infrastructure is elevated in the environment and thus lies above environmental obstacles to solar radiation. NIMS systems have exploited this and have been deployed with battery energy sources supplied solely by solar photovoltaic cell energy harvesting. Energy is transported as needed along the NIMS cable infrastructure by an articulated cable system. The NIMS system is deployed in a transect of length 70m and average height of 15m with a total area of over 1,000 m^2.

The experimental NIMS system operates with a linear speed range for node motion of 0.1 to 1 m/second. Thus, the time required to map an entire 1,000 m^2 transect with 0.1 m^2 resolution *with a simple 'raster' regular scanning schedule* will exceed 10^4 to 10^5 seconds. Phenomena that vary at a characteristic rate exceeding this scanning rate may not be accurately mapped. However, without knowledge of the spatial variability of measured phenomena, the sampling system must sample at high spatial rate in order to achieve low measurement distortion.

D. Event-Aware Adaptive Sampling Systems

Thus, as will be described below, the requirements for the measurement system to respond to variable environmental structure and to accurately map environmental variables at minimum time and resource cost has lead to the introduction of a NIMS architecture that combines mobile adaptive sampling with task allocation methods controlling node motion. These rely on fixed sensor networks as well as mobile elements to ensure that measurement is most efficient and then to focus exploration and mapping at the point in a transect for which change in environmental variables has occurred.

This paper first describes the Adaptive Sampling, Task Allocation, and also Event-Aware Adaptive Sampling methods and algorithms. A new Fidelity Driven Sampling adaptive method is introduced and tested with experimental data. Then, experimental methods including system hardware and software architecture are discussed. This includes description of a laboratory-scale NIMS system (NIMS-LS) employed for algorithm and verification of combined software and mobile embedded hardware system implementations. Finally, experimental results obtained with NIMS-LS demonstrating the marked benefits of combined adaptive sampling and task allocation are described. It is important to note that Related Work is discussed in each Section below.

II. Adaptive Sampling Algorithm

Sampling configurations (e.g., sample density) for embedded networked sensor systems are generally limited by

considerations related to the availability of communication channel resources or energy constraints. This has spawned a number of recursive estimation techniques that make decisions in a hierarchical fashion [5]. For example, local cluster heads accumulate information and propagate this according to a value metric. By introducing infrastructure-supported mobility, it is now possible to characterize environmental variable fields with much greater spatiotemporal density than was previously practical. While this enables new applications, it also creates new choices for estimators and for optimizing sampling. A new algorithm, Fidelity Driven Sampling will be described that exploits mobile sampling to first stratify the environment into regions requiring varying degrees of sample density, then samples in these regions while tracking estimated sampling error (fidelity). Fidelity Driven Sampling has the advantage over conventional raster (or other pre-planned sampling strategies) of actively seeking to minimize error without prior knowledge of the variable field.

A. Methodology

Optimal designs for estimators have been studied extensively in the statistics literature; see [6], [7], [8], [9]. These approaches often assume a parametric form for the estimate. Given a generic learning algorithm, the problem of optimal design is much harder. Optimal designs for simple kernel methods were studied by [10] where it was found that the best placement of points depends on the second derivative of the function; places with high curvature should have relatively more points than flat regions. [11] study designs for local linear smoothing, ultimately proposing a batch-sequential algorithm.

The NIMS adaptive sampling algorithms are designed with the goal of sequentially constructing a sampling pattern where at each step, samples are chosen that improve the estimate of the sensed field. [12] studies such sampling plans in the context of kernel smoothing. Points are added one at a time, chosen so as to reduce an estimate of the integrated squared error. [13], [14] study the same problem, but with neural networks as their estimators. They use the term "active learning" to describe the adaptive sampling process. [13] selects points to minimize an information criterion, while [14] studies the mean squared error. In each case, it is assumed the bias is negligible and attention is focused on the variance of the estimator. [14] also introduces the notion of path constraints on the samples, a topic that will be important in our ultimate deployment of adaptive sampling.

The Adaptive Sampling algorithm reported here, relies on a mean squared error estimation approach and an underlying learning algorithm based on a local linear smoother. Further, a variable bandwidth is assumed based on nearest neighbors. This algorithm, termed Fidelity Driven Sampling (FDS), attempts to reduce mean square error at each sampling point by adjusting point density and location. It will be further described and evaluated below.

```
predict-frame()
{
  //sequential design
  while   (there is a strata with error higher
           than threshold)
       extract strata with highest bias error
       add sampling points
       if  (current strata mean square error
           greater than threshold error)
         compute  vertical and horizontal mean
                  square error
         if  (horizontal bias error greater
             than vertical bias error)
             divide strata horizontally
             add stratum to the working queue
         else
             divide strata vertically
             add stratum to the working queue
         endif
       endif
    end while
  return local polynomial fit
}
```

Fig. 4. Pseudocode description of the Fidelity Driven Sampling algorithm.

Mean squared error can be decomposed into a bias component and a variance component. In our application of solar illumination mapping, measurement noise is a negligible fraction of the overall signal, and hence our greatest concern will be bias. In other applications, bias may not be dominant, and other estimation methods will be needed. In these other regimes, attention will likely focus on tasks other than field estimation; or if a snapshot of the field is desired, a longer learning process involving repeated measures in time will be required. This is all the subject of future work.

Throughout the sampling process, FDS maintains an estimate of the field being observed. In this paper, a local linear fitting routine is chosen, with its bandwidth varying to include a fixed number of nearest neighbors. Using this estimate, the FDS loop identifies regions or strata exhibiting a high degree of misfit. At each step in the sampling process, FDS adds points to that stratum with the largest error. In so doing, the FDS algorithm reduces the distance between neighbors and effectively lowers the bandwidth of the local linear fit within the stratum. The algorithm continues adding points to poor fitting strata until either an overall sample budget is exhausted or a desired fidelity limit is achieved.

Local linear fits have been chosen because these can tie the notion of sampling with the resolution of structures expected in the variable field. In principle, any nonparametric procedure could be employed including thin plate splines or other radial basis functions.

B. The Fidelity Driven Sampling System

Figure 4 shows pseudo code description of the Fidelity Driven Sampling adaptive algorithm. The NIMS robotic system continues adaptive sampling for the entire period of operation in the environmental mapping state. The FDS algorithm then calls the procedure predict-frame() which returns the estimate of the environmental variable field. The algorithm follows a procedure of stratifying a sampling region and according to observed measurements in the region and for each strata, adjusting the number of sampling points.

In addition, in the discussion below, the performance of Raster Scanning, Stratified Random Sampling, and the new Fidelity Driven Sampling algorithms are compared through simulation of each sampling method with actual experimental data. It is important to note that Fidelity Driven Sampling provides an autonomous system that seeks to assign sampling points to achieve a specified threshold error value (or a specified stratification rank). By estimating error magnitudes, Fidelity Driven Sampling can actively adapt to achieve a sampling fidelity objective. This differs fundamentally from raster scan methods that are not informed of residual errors. It is most important to note that Fidelity Driven Sampling, being adaptive, requires no prior knowledge of the environmental variable field characteristics and will rather report these characteristics. The successful demonstration of this is enclosed in Section VI.

This predict-frame() procedure starts by inserting a root stratum in a queue. Here, the root stratum corresponds to the entire transect - the entire region of study. predict-frame() then initiates a loop that extracts strata with highest product of mean square error and area. It determines the sampling points to be added to the strata and then the mobile sensor moves to visit those points and sample corresponding data. After sampling points in the strata, it performs a local linear kernel regression and reevaluates the estimate of the phenomenon. Error is computed in terms of the absolute difference between estimated and sampled values. If the computed error in the strata falls below a threshold, predict-frame() exits the inner loop and proceeds to examine more strata or otherwise divides the strata into horizontal or vertical substrata depending on which division leads to the greatest reduction in error.

Following Fidelity Driven Sampling operation (or raster scanning data acquisition) the returned variable field with its distribution of sample points was then supplied to a standard estimation algorithm (that performs an interpolation) and returns an environmental field map. This final result is referred to as a reconstruction of the variable field. Experimental results and evaluation of reconstructions will be discussed below for varying field characteristics.

III. Task Allocation Algorithm

Task Allocation (TA) is the problem of assigning available resources to tasks. There are two major subdivisions: offline and online. The offline TA is the problem of assigning resources to different tasks if certain information (e.g. the distribution of task arrival times, relative task priority) is known *a priori*. The assignment process proceeds offline. The offline TA problem, in its most general form, is equivalent to the conjunctive planning problem [15] which is NP-Complete.

Our focus here is on online task allocation. In online TA, all information about the tasks becomes available only upon task arrival. The assignment of resources to tasks must be computed in real time. It has been shown [16] that greedy algorithms provide good approximate solutions to online TA. It has also been shown [16] that in some cases the greedy online TA solution is within a bounded limit of the optimal solution obtained by offline TA. Following the model in [17], we think of task assignment occurring in *decision epochs*. A *decision epoch* is a period of time during which only the tasks which have arrived since the end of the previous epoch are considered for assignment. Increasing the *decision epoch* to infinity converts the online TA into the offline TA problem. We model the NIMS system as an online TA problem, since it is designed for real-life autonomous field applications in dynamic environments.

Our work is related to the body of work on the problem of online multi-robot task allocation (MRTA). For an overview and comparison of the key MRTA architectures see [18], which subdivides MRTA architectures into behavior-based and auction-based. For example, ALLIANCE [19] is a behavior-based architecture that considers all tasks for (re)assignment at every iteration based on robots' utility. Utility is computed by measures of acquiescence and impatience. Broadcast of Local Eligibility [20] is also a behavior-based approach, with fixed-priority tasks. For every task there exists a behavior capable of executing the task and estimating the utility of robot executing the task. Auction-based approaches include the M+ system [21] and Murdoch [22]. Both systems rely on the Contract Net Protocol (CNP) that puts available tasks for auction, and candidate robots make 'bids' that are their task-specific utility estimates. The highest bidder (i.e., the best-fit robot) wins a contract for the task and proceeds to execute it.

The proposed system differs from the above MRTA approaches. Our system relies on a static *network*, and communication, sensing and computation are distributed. The motivation for this system derives from the need to efficiently sample the entire environmental space. As has been discussed in the Introduction, it is impractical to deploy enough fixed sensors to achieve required sensing fidelity. As will be shown, the system combining fixed and

mobile nodes enables efficient sampling. TA becomes the primary driver of efficient data collection in this system, since it allows the user to select a portion of the environment for sampling, as opposed to sampling the entire environment. In addition, TA manages system resources, so that resources are not consumed unless assigned most effectively.

A. Methodology

The general online TA system functions in the following way. Suppose at a given decision epoch the system maintains a set of resources $R = \{r_1, ..., r_n\}$ and tasks $T = \{t_1, ..., t_k\}$. Tasks are prioritized based on a criterion C. C is an application dependent function and can combine such parameters as task arrival time, task importance, etc. A set of assignments $A = (l = min(n, k) : \{a_1, ..., a_l\})$ is computed as follows.

$$\forall_{a \in A} a = argmax_{j=(1,...,|R|)}(U(r_j, t)) \quad (1)$$

where t is the next unassigned task according to C and $U(r_j, t)$ is the j-th resource utility value for accomplishing t. The assigned resource and corresponding task are removed from R and T respectively, before the next assignment. The utility function is chosen to be application and resource dependent. In our model, once assigned, resources cannot be reallocated. After a resource has completed its task it becomes available for a new assignment. In the terminology of [23] we adopt a *commitment* strategy as opposed to *opportunism*.

The system consists of a mobile node suspended on a cable and a static sensor network. We assume that the network is predeployed where each node knows its location in a global coordinate system. The network monitors the environment for events of interest (motion, change in light intensity, etc). The problem then is to prioritize the events, and drive the mobile node to a vantage point from which a particular event is better observed. Once the node arrives, the local phenomenon is measured. In TA terminology, a robot is a *resource* and a detection by a sensor node of an event requiring perusal by a robot is a *task*.

Figure 5 shows two network topologies that we define - positioned on the ground (the 2D-case) and more generally, in the volume surrounding the transect (the 3D-case). In order for TA system to plan node's motion the goal points should lie in the transect plane. Hence, we project the nodes locations onto the transect plane. As a result we get a set of points on a line l (2D-case, Figure 5a) or a plane Π_r (3D-case, Figure 5c), both of which lie in the transect plane. In the 2D-case, l is the line where the transect plane intersects the ground plane. Since, the mobile node cannot move along that line, we translate l to a parallel line l_r on the transect. We define the projection function in the 2D-case $PROJ_{l_r}$ and 3D-case $PROJ_{\Pi_r}$.

Based on tasks projected locations TA divides the transect into *slices* (2D-case, Figure 5b), or generally *cells* (3D-case, Figure 5d). With every projected node k we associate a cell C_n.

Note that a 2D system is sometimes preferred because it is easier to setup in the field and for some applications a 2D perspective is enough. As an example, consider studying sunlight intensity shining through a forest canopy. In this case a sensor network with illumination detectors can be placed on the ground. Suppose node k discovered an interesting reading (say an abnormal light value). The TA system then would guide the robot towards the goal point on l_r computed by $PROJ_{l_r}$. The mobile node then can study appropriate *slice* C_k. The general 3D-case system is investigated in this paper.

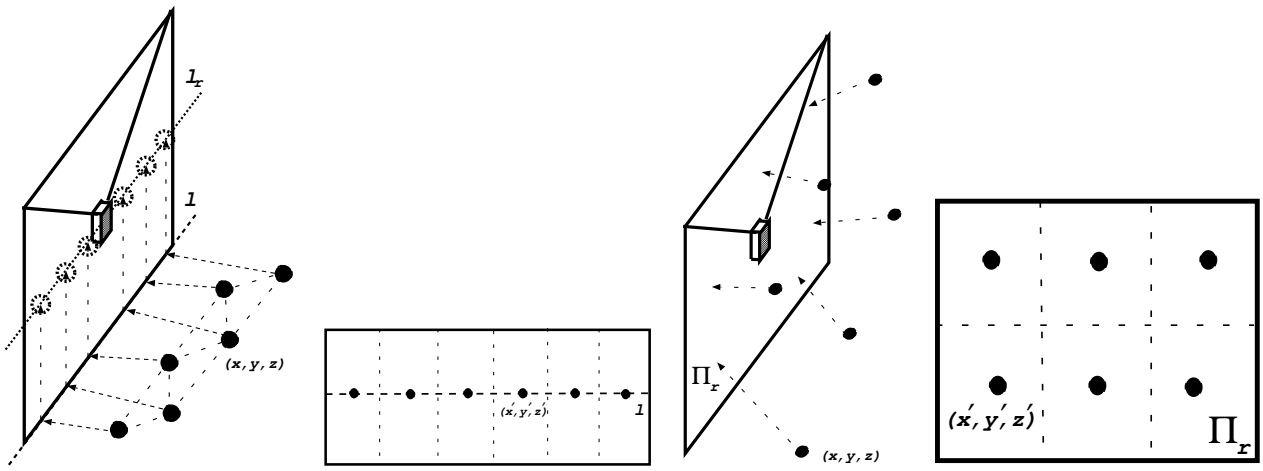
B. The Task Allocation System

Our system consists of two algorithms - one running on the robot and another within the network. First we describe the algorithm that runs on every static node. It consists of two parts - Task Generation and Task Management.

1) **Task generation**. If node's sensor reading is above a threshold a *task* is created and a notification message `NEW_TASK` is broadcasted. The notification message contains the notifying node's ID, location, sensor reading, and time stamp (time when the task was generated).
2) **Task management**. Each node maintains a set of currently active tasks T_a and non-active tasks T_{na}. If a node receives `NEW_TASK` message and the task is new (it is neither in T_a nor in T_{na}) then T_a is updated and the message is redirected to the network several times. As we discuss later, when the robot fulfills its assigned task it sends a `TASK_DONE` message containing the id of the node that generated the task and task's time stamp. If the task is in T_a, then it is removed from T_a, added to T_{na} and the message is redirected to SN several times.

Note that in practice the sizes of T_a and T_{na} are fixed (20 and 30 in our experiments). Both sets can be overwritten, so where is no overflow problem. At the same time, the size of T_a should be set with care to avoid loss of data about currently active tasks and potentially failing to propagate this data to the rest of the system.

Our system is a special case of the TA methodology described above - with only one resource (mobile node) for task assignment. Consider task assignment Equation 1. Since there is only one mobile node, the next task with highest priority (according to criterion C) is assigned to the mobile node, no matter what the mobile node's utility function might be. The task prioritization criterion selected here is C based on the time stamp associated with every task. The algorithm running on the mobile node is as follows.



(a) The 2D-case (b) The 2D-case projection onto the transect (c) The 3D-case (d) The 3D-case projection onto the transect

Fig. 5. Different SN topologies and corresponding projections onto the transect.

All incoming new tasks (specified by NEW_TASK message) are sorted according to the criterion C (tasks with smaller time stamp get priority - a FIFO policy) and stored in a set of currently active tasks T_a. When the mobile node becomes available for reassignment, the task of highest priority is extracted from T_a and assigned to it. Next, based on the task information the mobile node needs to compute a goal point.

If the task's position is p then the goal position will be $PROJ_{l_r}(p)$ in 2D-case and $PROJ_{\Pi_r}(p)$ in 3D-case (see Figure 5). After the robot completes its last task it sends a TASK_DONE message containing the id of the sensor node that generated the task and task's time stamp.

The TA algorithm computes projections of the static network onto the transect and separates the transect into slices (in the 2D-case) or more refined cells (in the 3D-case). This is the cornerstone for Event-Aware Adaptive Sampling Algorithm discussed next.

IV. Event-Aware Adaptive Sampling

The Adaptive Sampling approach we proposed in Section II (Fidelity Driven Sampling or FDS) is designed to capture static phenomena with an adjustable level of accuracy. On the other hand, consider a dynamic scene when the phenomenon to be observed changes spatially and temporally. If this change is faster than the time it takes adaptive sampling to complete, the algorithm would not obtain a correct result - the final "sensor picture" will consist of superimposed phenomena.

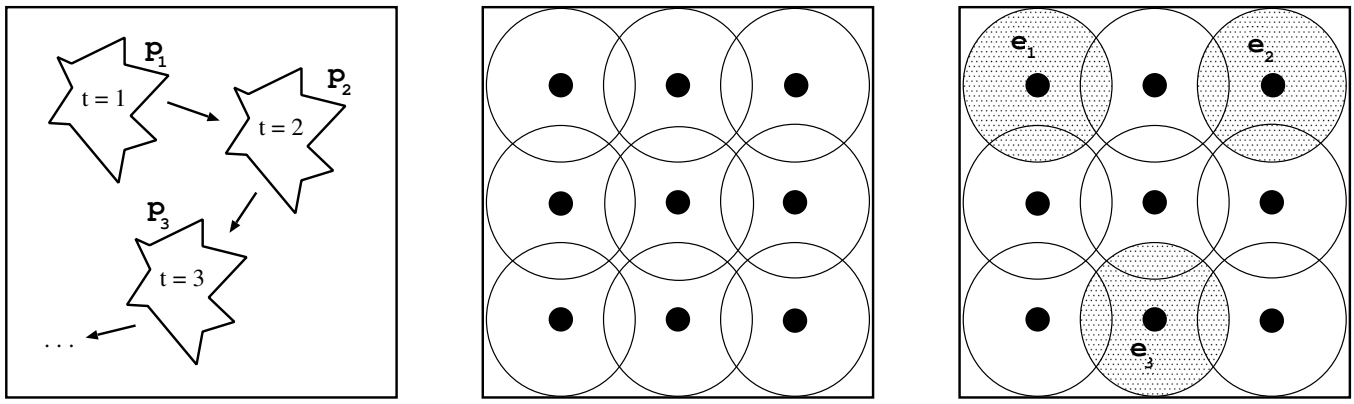
A. Methodology

Suppose the task is to observe a dynamic phenomenon P which changes rapidly in time (Figure 6a). Thus, P consists of $\{p_1, p_2, ...p_n\}$. One solution is to deploy many high fidelity sensors so that every point of the environment is sampled. This is impractical. Consider a hybrid approach where a large number of static low-fidelity sensors capable of detecting p_i are used in conjunction with the mobile NIMS node which carries the high fidelity sensor. The adaptive sampling algorithm can then focus on a portion of a transect containing p_i that TA system provides. As shown in Figure 6b, the sensor network effectively discretizes the environment, allowing to localize p_i and limit the adaptive sampling to a part of the transect. Note that multiple nodes can detect the same phenomenon p_i. For simplicity we assume that only the node with highest sensor reading of the phenomenon detects it. In principle, there are two ways to address that problem. One is to let the mobile node cluster tasks and then create a combined slice of the transect to run the adaptive sampling. Another way is to let the SN locally determine the cluster, and the 'leader' of the cluster will create a combined task.

B. Event-Aware Adaptive Sampling System

The following describes the Event-Aware Adaptive Sampling (EAAS) Algorithm:

1) TA module monitors the environment for new tasks;
2) If unassigned tasks set $T_a \neq 0$, TA assigns the robot to the next task in order, say task t;
3) TA determines the appropriate goal position and a corresponding transect cell as discussed in previous section;
4) TA delivers the robot to the computed goal position of the cell to sample;
5) TA sends request to the adaptive sampling system with (x, y, z) of the center of the scan and dimensions of the scanning cell;



(a) (b) (c)

Fig. 6. Dynamic phenomenon P and a static sensor network. a) Rapidly changing spatially and temporally phenomenon $P = \{p_1, p_2, ...p_n\}$, occurring forward in time; b) Static predeployed sensor network (SN), where each node is equipped with relatively cheap, low fidelity sensor capable of detecting p_i; c) SN detects p_1, p_2, p_3 and creates tasks;

6) TA system pauses, while monitoring the environment for new events;
7) The adaptive sampling scans designated area, returns to TA (sends AS_DONE_MSG);
8) TA sends `TASK_DONE` message into SN containing id of the node that generated the task and task's time stamp.
9) TA resumes. Repeat;

In summary, when a node detects a phenomenon (sensor reading reaches a predetermined threshold) it notifies the SN and the mobile node. We say that a task is created. The notification message contains the node ID, its location and the sensor reading. Given this information we can use the TA algorithm to assign robot to the task, navigate the robot to that task and start the adaptive sampling on a limited cell provided by TA. Figure 6c shows nodes that detected phenomena of Figure 6a and created corresponding tasks.

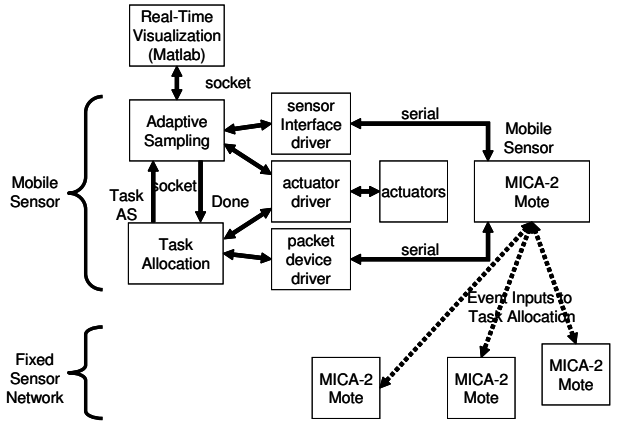


Fig. 7. The NIMS system architecture combines the adaptive sampling and Task Allocation (TA) algorithms providing interfaces to actuation and sensing. While the adaptive sampling relies on mobile sensor inputs, TA responds to events emanating from distributed fixed sensors. The adaptive sampling and TA algorithms are coordinated and hosted by the embedded Stargate platform shown in Figure 8

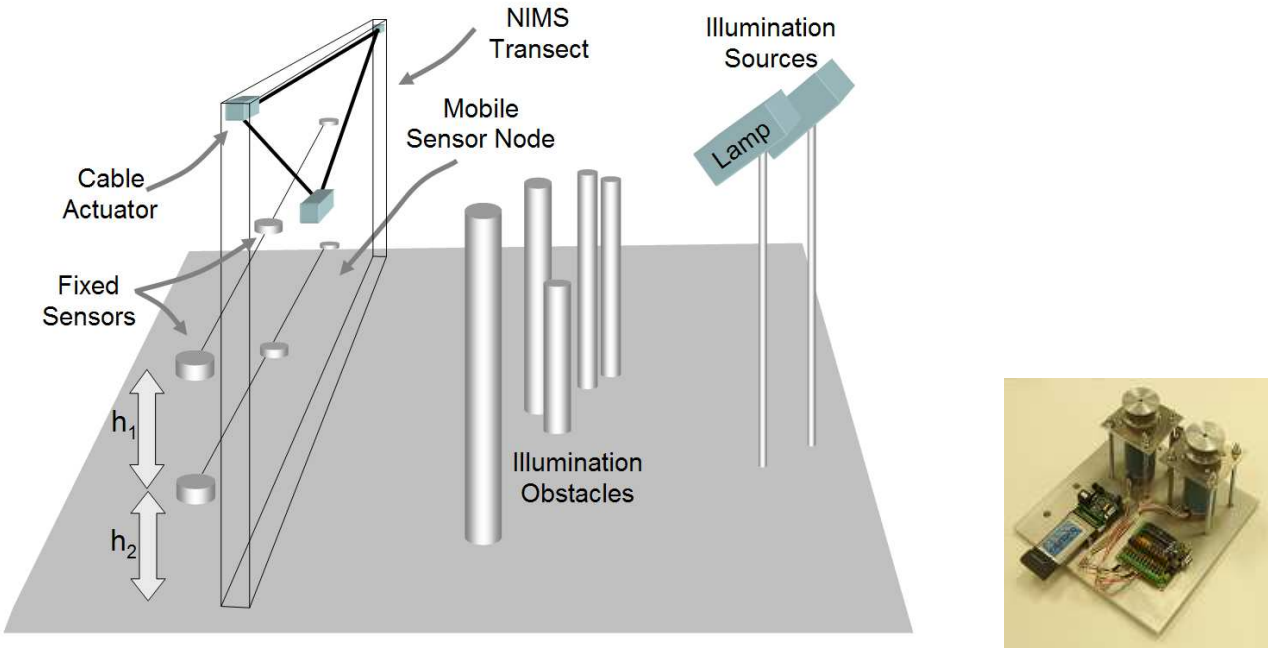
V. System Prototype and Experimental Setup

The combined Adaptive Sampling and Task Allocation methods have been introduced to both enable efficient measurement (for a specified accuracy estimate) while also enabling fast response to events. This section describes the enabling architecture and design methods.

A. System Hardware Prototype

As described in the Introduction, a NIMS system has recently been deployed in the field and characterization of forest canopy phenomena using NIMS methods may begin. However, the rapid development of NIMS algorithms and the verification of NIMS software, embedded hardware, and sensor systems benefit from characterization and testing with complete systems that operate in an indoor laboratory-scale facility where all inputs may be controlled. This laboratory-scale system, NIMS-LS, has been developed and is shown in Figure 7.

NIMS-LS includes a mobile node system suspended by a cable network (see Figure 8a) that is articulated by a "stepper" motor control system (see Figure 8b) controlled by a $Stargate^{TM}$ embedded node. The motor control system alternately winds and unwinds cable length from a pair of cable spools thereby causing a horizontal and vertical translation of the mobile node. Through proper calibration, this system provides less then 1 cm resolution for localization of the node at any point within the transect plane. While configurable in height and width, the transect used for the experiments reported here was 8m in length and 2.5m in height. Fixed sensor nodes were distributed



(a) (b)

Fig. 8. a) The NIMS-LS system employs a cable network to allow rapid deployment of a NIMS transect within an indoor environment. The cable system permits a sensor node to operate over the length and depth of a transect area. Software system interfaces are devised to enable applications to operate both on NIMS field-scale systems as well as NIMS-LS where development and verification may occur. In addition, reconfigurable lamp illumination sources and obstacles allow the designer to select a transect environmental variable distribution for the purposes of testing. Note that both mobile and fixed sensors appear in this architecture. Note that fixed sensors are distributed in the two dimensional transect plane as well. For the experimental characterization reported here, six MICA-2 mote sensors were deployed in two rows, equally spaced along the transect (of length 2m) and with the first row at a height of 80cm and second row at a height of 160cm as shown. b) The NIMS-LS system incorporates a standard mote sensor node for mobile measurement and an Intel $Stargate^{TM}$ platform for hosting of Adaptive Sampling and Task Allocation algorithms. The platform manages cable actuator control that, in turn, provides accurate node motion in the transect plane. This embedded platform, its motor systems, and cable spools for cable actuation are shown.

on the surface of the transect region and also elevated in the transect as well, as will be discussed below.

The mobile node is a standard wireless mote sensor system. In addition, this device is included in a wireless network with both the Stargate platform as well as the distributed fixed sensors.

B. System Software Prototype Architecture

The NIMS software architecture (as illustrated in Figure 7) is based on the Emstar system [24]. This has been selected for NIMS and provides the following benefits. First, Emstar provides a common set of embedded platform interfaces for multiple embedded platforms that compose NIMS field and NIMS-LS systems. This offers the benefit that applications developed for the NIMS-LS system may then be directly applied to NIMS field systems. Second, the Emstar event architecture provides the designer with robust methods for servicing unscheduled events and order of operations. The Adaptive Sampling and Task Allocation systems both benefit from this for implementation of their reactive nature. Third, the Emstar system provides a regular means for implementing the many complex device drivers that appear in NIMS electromechanical systems. Finally, Emstar has enabled NIMS emulation that includes rapid verification of mobile sensing algorithms (on standard x86 workstation platforms) in preparation for testing on the NIMS-LS platform.

The Adaptive Sampling and Task Allocation algorithms operate on the Stargate platform and communicate over socket interfaces as shown in Figure 7 Both Adaptive Sampling and Task Allocation exploit Emstar device drivers. The degree to which this Emstar-based architecture accommodates diverse applications is illustrated by this example. Here, the Adaptive Sampling algorithm (with its interfaces to motor control systems) is developed in C and C++. However, the Task Allocation algorithm is implemented in Java. The NIMS-LS architecture has incorporated this diversity and enables robust operation of the combined complex and event-ware Adaptive Sampling and Task Allocation processes.

C. Experimental Setup

The environmental variable field is created through use of an optical lamp system that illuminates the transect.

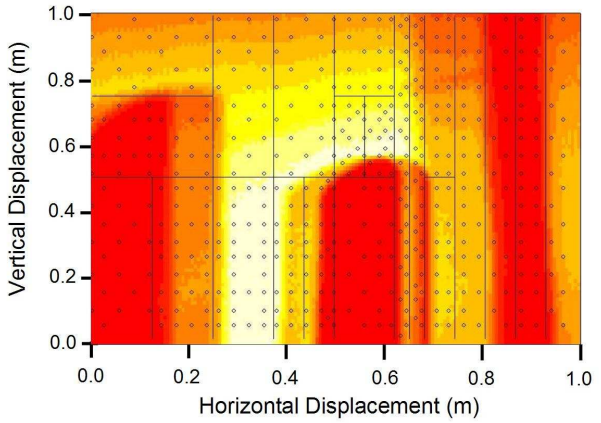


Fig. 9. An environmental field map for a complex variable field with superimposed sampling points as autonomously selected by Fidelity Driven Sampling. The image grey level indicates illumination intensity variation (varying by a factor of 5.7 from the darkest to lightest regions).

Just as in the natural environment, photodiode sensors are employed for light intensity measurement. In addition, cylindrical obstacles are available and are placed in the environment to create a variable field for test purposes. For the experiments reported here, the variable field was selected to follow that observed in the natural environment.

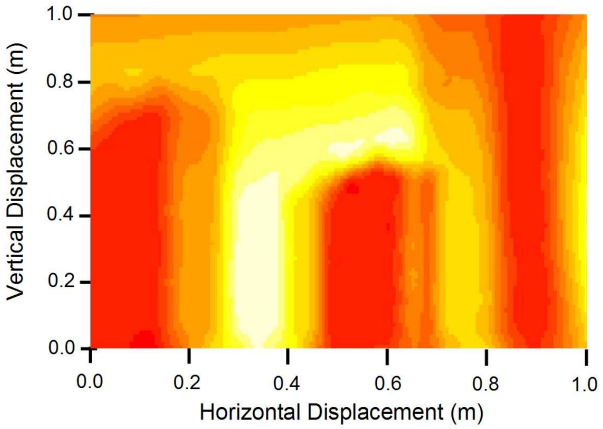


Fig. 10. The reconstructed variable field derived from the Fidelity Driven Sampling points of Figure 9

VI. Results for Adaptive Sampling

As described in Section II, Fidelity Driven Sampling exploits mobile sensing capabilities to explore the variable field, stratify this into regions of greatest required sample density, and then sample in these regions adaptively to minimize estimated sampling error. Fidelity Driven Sampling operates in an iterative architecture seeking to reach

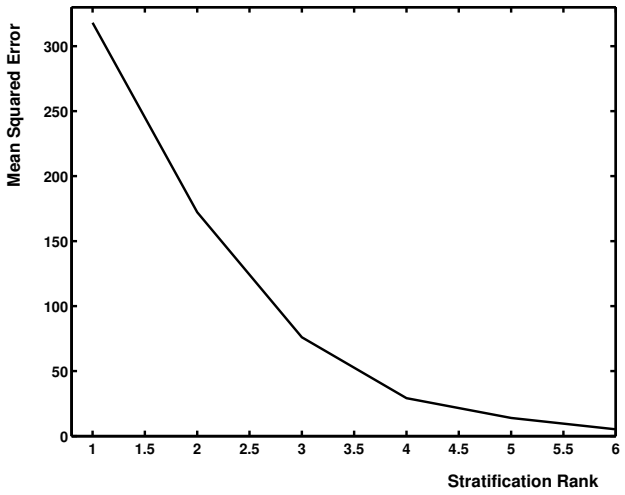


Fig. 11. Mean Squared Error for reconstructions of the environmental field of Figure 9 as a function of stratification level. Note that unlike pre-planned raster scanning or related sampling strategies, Fidelity Driven Sampling may proceed autonomously reach a specified error threshold.

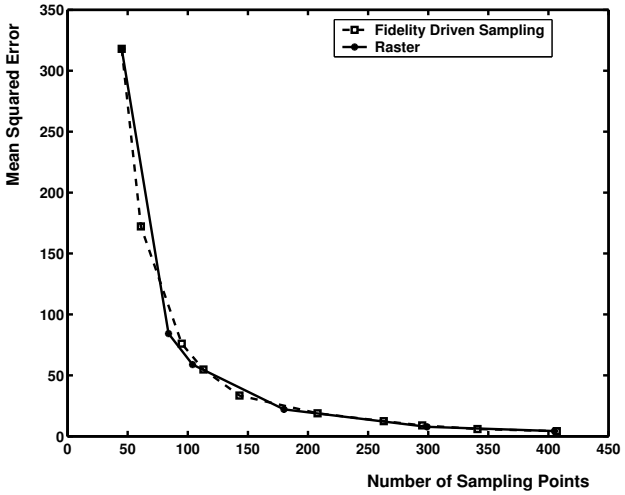


Fig. 12. Mean Squared Error for reconstructions of the environmental field of Figure 9 as a function of sampling points comparing Fidelity Driven with raster sampling. Note that while raster sampling may exhibit a large variation in error with choice of sample density, Fidelity Driven sampling autonomously seeks a low value.

a desired threshold error. While multiple error threshold policies may be applied, two are explored here: 1) an error threshold defined as a maximum tolerated mean squared error estimate across the entire environmental field, and 2) a maximum allowed stratification rank. Adjustment of these thresholds permit the mobile sensor node to return an environmental field map with a specified estimated fidelity without the requirement for any prior knowledge of the field characteristics. An inability to reach a specified estimated error (within a given time or rank level limit) will

be reported by the Fidelity Driven Sampling algorithm. This then provides the user with yet further assurance of proper sampling and confidence in returned data.

Fidelity Driven Sampling is evaluated here by subjecting the algorithm to environmental variable fields having two extremes in their "curvature" characteristics. For one limit, the environmental variable field was created by placing many obstacles in the illumination field (see Figure 8) to emulate the characteristically most complex patterns observed in the natural environment. In addition, the algorithm was also subjected to an environmental variable field that showed low curvature created by casting only diffuse shadowing on the transect. This latter case is characteristically similar to the least complex fields observed under clear forest canopy structure.

The performance of the Fidelity Driven Sampling algorithm was evaluated by allowing the algorithm to autonomously operate and return a sample distribution. This distribution then was supplied to the estimator to return a reconstructed variable map. Finally, this map was compared with the actual measured data obtained by exhaustively moving the node at his highest resolution through the variable field. This returned a "ground truth" map of the scene.

The results of Fidelity Driven Sampling were then compared with conventional raster scanning data acquisition that performs the role of a conventional pre-planned and non-adaptive sampling strategy.

Fidelity Driven Sampling algorithm shows a value of Mean Squared Error for its reconstruction compared to ground truth that meets or is superior to that of raster scanning. However, this must be achieved without pre-planning and must be independent of the nature of the field characteristic. We examined both the returned reconstruction as well as its mean of squared error over the entire transect.

It is important that Fidelity Driven Sampling shows a monotonic decrease in mean squared error for an increase in stratification rank level. We compared the result of FDS with raster scanning for both rough and smooth phenomena.

A. *Fidelity Driven Sampling vs. Rough Phenomena*

Figure 9 shows a map of both ground truth and the positions of both strata and actual sample points selected by Fidelity Driven Sampling during an experimental session. Note that the sample point density increases in regions of greatest field curvature. This experimental result was captured as the Field Driven Sampling system passed through stratification rank 5 and had selected 489 sample points within the transect. Figure 10 shows the reconstruction resulting from this. Close agreement in field shapes is observed.

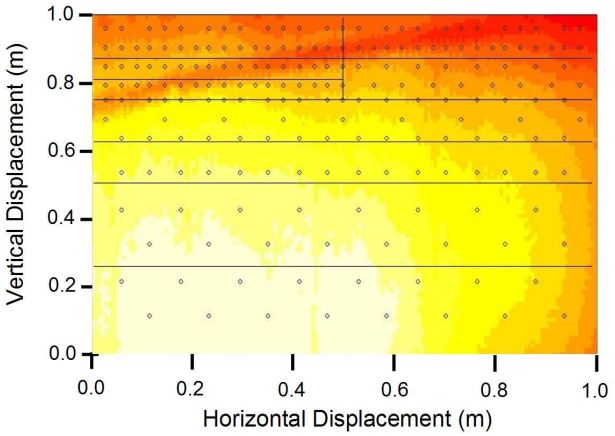


Fig. 13. n environmental field map for a low curvature variable field with superimposed sampling points as autonomously selected by Fidelity Driven Sampling. The image grey level indicates illumination intensity variation (varying by a factor of 1.5 from the darkest to lightest regions).

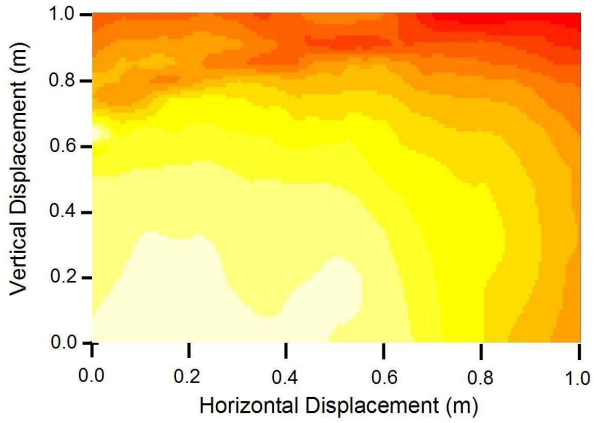


Fig. 14. The reconstructed variable field derived from the Fidelity Driven Sampling points of Figure 13

A test of the performance of this approach is shown in Figure 11. Note that the actual mean squared error (computed between the reconstruction and ground truth across the transect) reduces with increasing stratification rank. Also, Figure 12, shows the dependence of mean squared error on sample density for Fidelity Driven Sampling (Adaptive) and raster scanning (Raster) methods.

B. *Fidelity Driven Sampling vs. Smooth Phenomena*

The sample distribution results and reconstruction results for a rank level of 5 are shown in Figures 13 and 14 for an environmental field showing dramatically less curvature than that of Figure 9. A comparison of mean squared error performance between Fidelity Driven and raster scanning is shown in Figure 14. Note that for this reduced curvature, not only does Fidelity Driven Sampling

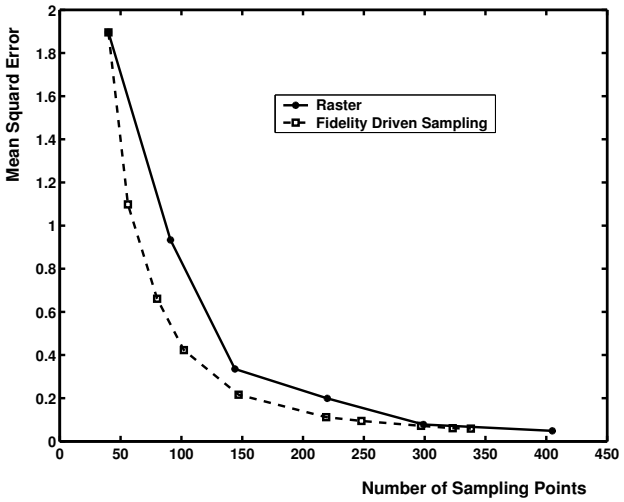


Fig. 15. Mean Squared Error for reconstructions of the environmental field of Figure 13 as a function of sampling points comparing Fidelity Driven with raster sampling.

converge to a specified mean squared error, it is equal or superior to raster scanning in efficiency with respect to numbers of sample points.

The combination of these tests provides a successful evaluation of the performance of Fidelity Driven Sampling for autonomously adjusting sample density through appropriate motion control and sampling of the mobile sensing system. As has been discussed this sampling method seeks to establish a reconstructed variable field with a specified maximum sampling error without requiring prior knowledge or planning.

VII. Results for Event-Aware Adaptive Sampling

Both Task Allocation and Event-Aware Adaptive Sampling use a notion of a task. In the following experiments we will compare the cumulative task *OnTime* across all tasks, over the duration of every experiment. Each task's *OnTime* is computed as the difference between the time the task was turned off by a Robot (`TASK_DONE` message is sent) and the time the task was detected by one of the nodes of the network.

As shown on Figure 8, a network of 6 Mica2 Motes was predeployed in a test environment with predetermined coordinates. We use the general 3D topology. Hence, by knowing nodes locations and computing nodes' projections onto the transect plane, the TA algorithm produces a subdivision of the transect similar to Figure 5b.

A. Task Allocation vs. Raster Scan

Following the same approach as described for characterization of the Fidelity Driven Sampling system component, experiments were conducted comparing Task Allocation(TA) methods with conventional Raster Scan methods. The Raster Scan method scans every point of the transect with a specified resolution. When the Raster Scan reaches the location of an event it clears it by sending `TASK_DONE` message. Raster Scan method proved to be prohibitively low in performance. In particular, experimental results showed that at the maximum NIMS-LS spatial resolution of 1 cm, with a sampling dwell time of 1 second at each location, *OnTime* results were dramatically inferior to TA methods. Raster Scan method was also characterized at reduced spatial resolution of 5cm with a corresponding improvement in response time. This however, is still inferior to TA algorithm described in this paper.

In this experiment an artificial event is first generated on a remote server. Then the server sends an event message to the node designated for task generation and the node proceeds as if this event was detected by the node's sensor. For this experiment, schedules of 3, 5, 7, 10 and 20 events were drawn (in time) from a Uniform distribution to arrive within 600 seconds, with uniformly distributed nodes that detected the event. Note that for actual applications we do not expect to receive/process more than 1 - 10 events in 10 minutes on average. Hence the case of 20 events shows the behavior of the system at the limit.

Figure 16 shows experimental results comparing *OnTime* performance of TA and Raster Scan. The number of events varies between 3 and 20. Both algorithms were evaluated from 3 different starting positions of the mobile node on the transect (drawn from a Uniform distribution). The results were averaged. As can be seen from the graph, TA performs 9-22 times better on the entire interval of 3-20 events. Note also that TA is stable, as indicated by error bars, and hence is favored for use in this application since it provides reduced bounds on system run time over Raster Scan method.

In addition to response time comparison, it is also important to compare mobility requirements for TA and Raster Scan methods. Specifically, the use of mobility requires energy. Thus, this can be computed and compared for each method. Now, when the density of the events is low, the TA algorithm enables the mobile node system to remain in a static position for extended periods - "in between events". This occurs when it has serviced all events that have arrived and is awaiting new events. Raster Scan, however, forces the robot to move constantly. Hence, this method will consume far greater energy and mobility resources than TA. A measure of energy for mobility is determined for the purposes of comparison by computing the total time of mobile node motion. Figure 17 shows comparison of energy consumption in units of time-in-motion between TA and Raster Scan. As expected, TA outperforms Raster Scan significantly. However as the number of events increases to infinity, the TA should approach Raster Scan energy consumption. Also note, that

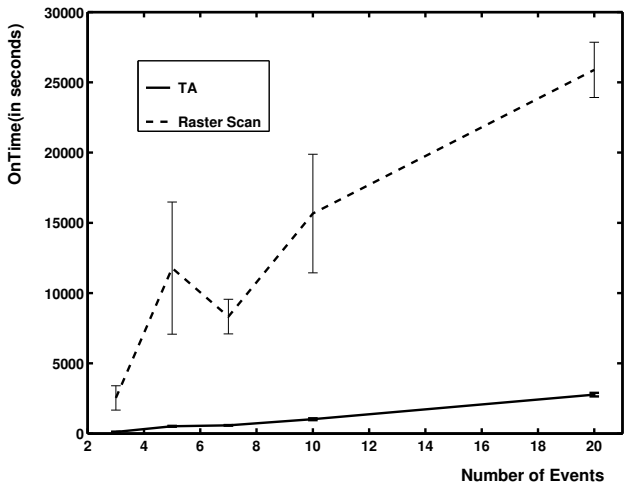


Fig. 16. Comparison of event OnTime between TA and Raster Scan. Number of events varies between 3 and 20.

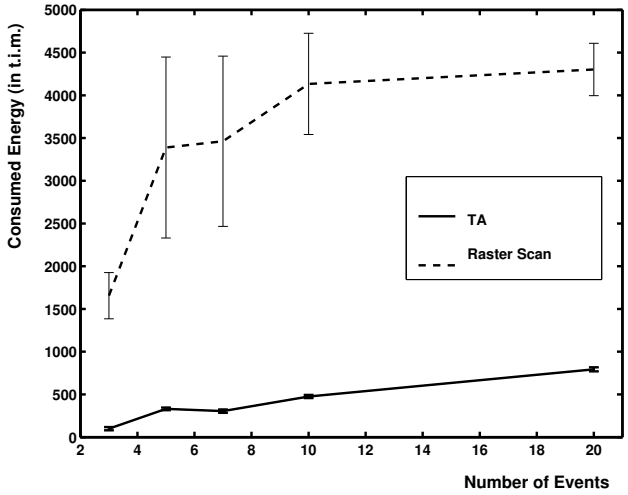


Fig. 17. Comparison of energy consumption in units of time-in-motion (t.i.m.) between TA and Raster Scan. Number of events varies between 3 and 20.

on interval [5, 20] the slope of the Raster Scan curve is very small and the energy consumption is insensitive to event arrival rate.

B. Event-Aware Adaptive Sampling

The introduction of Event-Aware Adaptive Sampling produces a potential dramatic improvement in event response time. Thus, in this section the results of measurement of task OnTime is described for the system using the Event-Aware Adaptive Sampling (EAAS) algorithm. A previous investigation has evaluated an algorithm approach intended to enable mobile sensor nodes to perform an adaptive samplinnog [25]. This was evaluated on data from a mobile sensor system that was acquired using a regular raster scan. In contrast, the system architecture reported here allows adaptive sampling algorithms to be hosted directly on mobile embedded systems. In addition, this system architecture enables cooperation between adaptive sampling and Task Allocation algorithm. This is demonstrated through direct experimental evaluation with the NIMS-LS system. The results for EAAS operation combining task allocation and adaptive sampling are compared directly with results for the adaptive sampling system operating alone.

It is important to note that adaptive sampling does not provide an accurate sampling of the scene if the frequency of events is such that events arrive at a rate greater than the time required for adaptive sampling to complete. But if the event frequency will be sufficiently low then on average every task *OnTime* will be approximately the same. Hence, it suffices to compare the results for *OnTime* of only one task.

This experimental characterization was performed using light intensity variations to induce events (exactly as occurs in natural environment conditions.) As shown in Figure 8, the mobile and fixed nodes form a network. Also, each node is equipped with a light intensity sensor as deployed on the mobile node. Here, a node generates an event when the sensor sampled value exceeds a threshold.

The task *OnTime* performance of adaptive sampling is 20524 seconds and 577 seconds for EAAS. Hence EAAS significantly outperforms adaptive sampling (with an improvement by a factor greater than 35). This dramatic advantage of EAAS results from its ability to determine the location of the task, deliver the mobile node to the proper location, and then reduce the scan area requirements for the adaptive sampling module by focusing attention on the proper event region.

VIII. Summary and Future Work

Sensor network systems are now being applied in critical science and engineering applications in complex environments. Such environmental field characterizations, confronts the challenge of spatiotemporal evolution of obstacles in the sensing environment, introducing an unknown level of measurement distortion. This paper describes a new architecture that augments fixed sensor networks with infrastructure-supported mobile nodes. This architecture incorporates systems that exploit adaptive mobility to actively explore environments and determine sampling point distribution, achieving a specified level of sensing fidelity. The coordination of fixed and mobile nodes also calls for new task allocation algorithms and associated systems that direct mobile node resources to events. This paper describes the integration of a Fidelity Driven Adaptive Sampling method with new Task Allocation algorithms and systems. Experimental measurements demonstrate that this combined system offers dramatically

improved spatiotemporal coverage over that of either fixed or mobile sensors alone.

Future work will both develop new Fidelity Driven sampling and exploration systems and also apply these systems more broadly to field biology and civil engineering environmental monitoring needs. Sensing diversity will also be introduced to enhance Fidelity Driven Sampling with measurements made available from fixed and mobile sensors of diverse modes and locations. It is anticipated that this will permit Bayesian-style updating or learning that can draw on possibly coarse or noisy auxiliary measurements when forming an estimate of the variable field.

Finally, New architectural features will appear that include 'sensor strands' consisting of fixed nodes suspended from the infrastructure in the three dimensional environment and emplaced and reconfigured by mobile NIMS node systems. Multiple NIMS mobile node transects will also be deployed in large and varied environments and transect types. This will lead to investigation and development of new utility functions for task allocation that coordinate fixed node networks and multiple mobile assets for further advances in sensing fidelity for complex environmental field characterization.

IX. Acknowledgments

Left out for anonymity.

X. References

[1] D. Estrin, G.J. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *ICASSP 2001*, Salt Lake City, USA, 2001.

[2] C. M. P. Ozanne, D. Anhuf, S. L. Boulter, M. Keller, R. L. Kitching, C. Korner, F. C. Meinzer, and A. W. Mitchell, "Biodiversity meets the atmosphere: A global view of forest canopies," *Science*, vol. 301, pp. 183–186, 2003.

[3] *Radiation and Light Measurements*, chapter 6, pp. 97–116, Physiological Ecology - Field Methods and Instrumentation. Chapman & Hall, London, U.K.

[4] ," http://www.jamesreserve.edu/.

[5] R. Nowak, U. Mitra, and R. Willett, "Estimating inhomogeneous fields using wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, 2004.

[6] J. Kiefer, "Optimal experimental design," in *J. Roy. Statist. Soc., Ser. B, 21, 272-319*, 1959.

[7] V. V. Fedorov, *Theory of Optimal Experiments*, New York: Academic Press, 1972.

[8] S. D. Silvey, *Optimal Design*, London: Chapman & Hall, 1980.

[9] R. Pukelsheim, *Optimal Design of Experiments*, New York: Wiley, 1993.

[10] H Müller, "Optimal designs for nonparametric kernel regression," in *Statistics and Probability Letter*, 1984.

[11] M-Y Cheng, P. Hall, and M. Titterington, "Optimal design for curve estimation by local linear smoothing," in *Bernoulli 4(1), 3-14*, 1998.

[12] J. Faraway and D. Park, "Sequential design for response curve estimation," 1998, pp. 9, 155–164.

[13] D.J.C. MacKay, "Information-based objective functions for active data selection," in *Neural Comput.*, 1992, pp. 4, 590–604.

[14] D. A. Cohn, "Neural netowrk exploration using optimal design," in *MIT AI Lab Memo No. 1491.*, 1994.

[15] D. Chapman, "Planning for conjunctive goals," *Artificial Intelligence*, vol. 32, pp. 333–377, 1987.

[16] Bala Kalyanasundaram and Kirk Pruhs, "Online Weighted Matching," *J. of Algorithms*, vol. 14, pp. 478–488, 1993.

[17] M. A. Batalin and G. S. Sukhatme, "Using a sensor network for distributed multi-robot task allocation," in *To appear in Proc. of IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, USA, 2004.

[18] B. Gerkey and M. J. Mataric, "Multi-robot task allocation: Analyzing the complexity and optimality of key architectures," in *To appear in Proceedings of the IEEE International Conference on Robotics an Automation (ICRA03)*, Taipei, Taiwan, 2003, pp. 3862–3868.

[19] L. E. Parker, "Alliance: An architecture for fault-tolerant multi-robot cooperation.," in *IEEE Transactions on Robotics and Automation*, 1998, vol. 14, pp. 220–240.

[20] B. B. Werger and M. J. Mataric, *Distributed Autonomous Robotic Systems 4*, chapter Broadcast of Local Eligibility for Multi-Target Observation, pp. 347–356, Springer-Verlag, 2000.

[21] S. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement.," in *Proc. of IEEE International Conferenceon Robotics and Automation (ICRA)*, 2000, pp. 293–298.

[22] B. P. Gerkey and M. J. Mataric, "Sold!: Auction methods for multi-robot coordination.," in *IEEE Transactions on Robotics and Automation*, 2002, vol. 18, pp. 758–768.

[23] E. H. Ostergard, M. J. Mataric, and G. S. Sukhatme, "Distributed multi-robot task allocation for emergency handling.," in *In Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2001, pp. 821–826.

[24] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, "Emstar: a software en-

vironment for developing and deploying wireless sensor networks," in *to appear in Proceedings of USENIX 04*.

[25] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava, "Adaptive sampling for environmental robotics," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, New Orleans, LA, 2004.