The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Multiscale Sensing with Stochastic Modeling

Diane Budzik, Amarjeet Singh, Maxim A. Batalin and William J. Kaiser
Center for Embedded Networked Sensing, University of California, Los Angeles
Email: dbudzik@ee.ucla.edu

Abstract— **Many sensing applications require monitoring phenomena with complex spatio-temporal dynamics spread over large spatial domains. Efficient monitoring of such phenomena would require an impractically large number of static sensors; therefore, actuated sensing - mobile robots carrying sensors - is required. Path planning for these robots, i.e., deciding on a subset of locations to observe, is critical for high fidelity monitoring of expansive areas with complex dynamics. We propose *MUST* - a *MU*ltiscale approach with *ST*ochastic modeling. *MUST* is a hierarchical approach that models the phenomena as a stochastic Gaussian Process that is exploited to select a near-optimal subset of observation locations. We discuss in detail our proposed algorithm for the application of monitoring light intensity in a forest understory. We performed extensive empirical evaluations both in simulation using field data and on an actual cabled robotic system to validate the effectiveness of our proposed algorithm.**

I. Introduction

Many diverse environmental monitoring applications involve observing phenomena that exhibit both spatial and temporal dynamics spread over large spatial domains. Solar light radiation, carbon dioxide flux, algal blooms, and nitrate concentration in a watershed are a few examples of such phenomena [1], [2]. Consider the significant role that tropical forests and the plants growing in their understory play in carbon dioxide flux. A detailed understanding of plant growth and photosynthesis in this environment is needed to model how deforestation and forest degradation affect carbon emissions (degradation estimates range anywhere from 7-30% [3]). Accurate monitoring of solar radiation distribution in forests will lead to a better understanding of plant growth and the role forest degradation plays in carbon dioxide flux.

Measuring complex space and time dynamics, such as solar radiation, with high fidelity in large environments, such as forests, with only static sensors would require an impractically large number of sensors to be distributed across the complete spatial extent of the observed environment. Therefore, actuated sensing - mobile robots carrying the required sensors - is needed to accurately measure complex spatio-temporal dynamics over large spatial domains. Monitoring the complete environment with a limited number of actuated sensors will, however, increase the sensing delay, which may result in lower sensing fidelity. Planning robot paths, i.e., adaptively selecting only a small subset of locations to observe while still efficiently predicting at the unobserved locations, is critical to reduce latency and still provide high fidelity sampling. Several path planning algorithms have been proposed in the literature [4]–[7] that reduce the number of locations to observe while still achieving high fidelity.

For example, Rahimi et al. [4] proposed a multi-step approach that varies observation densities at each step. In the first step, a mobile robot performs a coarse scan of the complete environment to extract regions of high phenomenon variability. Then the selected regions are observed with a higher density to improve the overall sampling fidelity. Such adaptive sampling algorithms are known to perform well in cases where the observed phenomena are not highly dynamic in both space and time. However, the large latency involved in extracting these regions of interest makes such algorithms unsuited for observing rapidly changing phenomena. This large latency is mainly a result of the initial coarse scan that is performed over the entire environment. If the phenomena is only spatially dynamic, this latency is not prohibitive. However, as the latency increases with the size of the environment, adaptive sampling does not scale well for temporally dynamic phenomena. Singh et al. [5] proposed a two-tier approach to reduce this large latency where the first tier uses a static low fidelity, high spatial coverage sensor providing real-time "global" information about the environment. The second tier, using actuated sensing, exploits the information from the first tier to perform guided sampling in the regions of interest. However, the second tier only performs uniform sampling within the regions of interest.

Krause et al. [8] considered the sensor placement problem, where a subset $\mathcal{A}$ of k locations is selected from all possible observation locations $\mathcal{V}$, ($\mathcal{A} \subseteq \mathcal{V}$), in order to maximize the collected information about phenomena distribution. By exploiting the intuitive "diminishing returns" property - the more locations that are already observed, the less information we will gain by observing a new location - they proved that greedily selecting locations based on this criterion is near-optimal. Singh et al. [7] extended their approach to provide an efficient path planning algorithm for adaptively sampling the observed environment at only a small subset of locations while also including the path cost. We also gave a strong approximation guarantee for the proposed algorithm. The time complexity of their algorithm, however, may restrict its utility for very large spatial domains.

Motivated by the effectiveness of the spatial decomposition algorithm and by the greedy selection of observation locations within the selected cells (regions of interest) [7], we propose *MUST* - a *MU*ltiscale approach with *ST*ochastic modeling. *MUST* extends the two-tier multiscale algorithm proposed in [5] by using a stochastic modeling approach together with greedy observation selection from [8]. Bringing these approaches together allows *MUST* to effectively address both the challenge of observing complex phenomena in

a large spatial domain and the challenge of reducing latency in order to capture temporal variations. To accomplish this, *MUST* uses the information from the first tier "global" sensor to learn a stochastic *Gaussian Process* (GP) model. Next, the path planning algorithm exploits the GP model by greedily selecting a small subset of locations to observe with the second tier sensor in the regions of interest, which are also selected by the first tier. This efficient path planning algorithm is easily extensible to the multiple robot setting (c.f. Section III-A). We empirically demonstrate the effectiveness of our approach for the application of monitoring light intensity in the forest understory. We performed extensive empirical evaluations both in simulation using data collected in the field as well as on an actual cabled robotic system - Networked Infomechanical System for Planar actuation (NIMS-PL) [9]. Results indicate that the hybrid approach of *MUST* considerably and consistently outperforms a provably efficient path planning algorithm for light intensity monitoring.

In Section II we briefly discuss the Gaussian Process modeling approach and the corresponding properties that provide a near-optimal approximation guarantee for greedy observation selection within a task. Section III outlines two path planning algorithms: 1) our proposed *MUST* algorithm; and 2) the *OP heuristic* - an efficient path planning algorithm shown to perform well for the application of light intensity monitoring [9]. We discuss extensive empirical evaluations in Section IV, including experiments performed using a cabled robotic system, NIMS-PL.

II. Gaussian Process Modeling

Let us now introduce some notation and review Gaussian Process models for regression [10]. We are given a set of observations, $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, consisting of N input locations, $\mathbf{x}_i \in \mathbb{R}^D$, and the corresponding observed values, $y_i \in \mathbb{R}$. A Gaussian process model places a multivariate Gaussian distribution over the space of function variables, $f(\mathbf{x})$, mapping the input to the output space, i.e., $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$, where $m(\mathbf{x})$ specifies a mean function and $k(\mathbf{x}, \mathbf{x}')$ specifies a covariance function (also called the kernel).

Observations in the environment are typically noisy samples of the underlying model. This noise is incorporated by assuming $y = f(\mathbf{x}) + \epsilon$, where ϵ is the zero mean Gaussian noise with variance σ^2. The set of locations, model output, and observations at the observed locations are represented by $(X, \mathbf{f}, \mathbf{y}) = (\{\mathbf{x}_i\}, \{f_i\}, \{y_i\})_{i=1}^N$, and unobserved locations are represented as $(X_*, \mathbf{f}_*, \mathbf{y}_*) = (\{\mathbf{x}_{*,i}\}, \{f_{*,i}\}, \{y_{*,i}\})_{i=1}^N$. Using the GP model learned from the observations, the objective is to compute the predictive distribution f_* at the unobserved locations X_*. Conditioning on the observations, the predictive distribution at the unobserved locations can be obtained by:

$$p\left(f_* \mid X_*, X, \mathbf{y}\right) = \mathcal{N}\left(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*\right), \tag{1}$$

$$\begin{aligned} \boldsymbol{\mu}_* &= K\left(X_*, X\right)\left[K\left(X, X\right) + \sigma^2 I\right]^{-1}\mathbf{y}, \\ \boldsymbol{\Sigma}_* &= K\left(X_*, X_*\right) + \sigma^2 I - \\ &\quad K\left(X_*, X\right)\left[K\left(X, X\right) + \sigma^2 I\right]^{-1} K\left(X, X_*\right), \end{aligned}$$

where $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}_*$ represent the conditional mean and covariance, respectively, at the unobserved locations.

Informative observation locations: Intuitively, the most informative observation locations result in the greatest reduction of uncertainty at unobserved locations. A natural definition of uncertainty is the conditional entropy of unobserved locations $\mathcal{V} \setminus \mathcal{A}$ after placing sensors at $\mathcal{A}$ locations,

$$H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}|\mathcal{X}_{\mathcal{A}}) = -\int p(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}}, \mathbf{x}_{\mathcal{A}}) \log p(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} \,|\, \mathbf{x}_{\mathcal{A}}) d\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} d\mathbf{x}_{\mathcal{A}},$$

where $\mathcal{X}_{\mathcal{A}}$ and $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ refer to sets of random variables at locations $\mathcal{A}$ and $\mathcal{V}\setminus\mathcal{A}$. The differential entropy for a Gaussian random variable, $\mathcal{X}_y$, conditioned on some set of variables, $\mathcal{X}_{\mathcal{A}}$, is a monotonic function of its variance:

$$\begin{aligned} H(\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}) &= \tfrac{1}{2}\log(2\pi e \sigma^2_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}}) \\ &= \tfrac{1}{2}\log\sigma^2_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}} + \tfrac{1}{2}(\log(2\pi)+1) \end{aligned} \tag{2}$$

Thus for the unobserved locations, we can compute the entropy in closed form using the conditional covariance matrix computed in Eq. (1). Instead of directly using the entropy as the information criterion, we use the *mutual information* (MI) criterion defined in [11] for representing sensing quality. For a set of locations $\mathcal{A}$, the MI criterion is defined as:

$$\mathrm{I}(\mathcal{A}) \equiv \mathrm{MI}(\mathcal{A}) \equiv H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}) - H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}), \tag{3}$$

where $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}})$ is the entropy of the unobserved locations $\mathcal{V}\setminus\mathcal{A}$, and $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}})$ is the conditional entropy of locations $\mathcal{V}\setminus\mathcal{A}$ after sensing at $\mathcal{A}$ locations. The effectiveness of mutual information in selecting informative sensing locations was studied in [8].

Submodularity: A key observation allowing us to obtain an efficient algorithm is that mutual information satisfies the following diminishing returns property [8]: The more locations we have already observed, the less information we will gain by observing a new location. This intuition is formalized by the concept of *submodularity*: A function f is *submodular* [12] if:

$$f(\mathcal{A}\cup s) - f(\mathcal{A}) \geq f(\mathcal{B}\cup s) - f(\mathcal{B}); \forall \mathcal{A}\subseteq\mathcal{B}\subseteq\mathcal{V} \text{ and } s\in\mathcal{V}\setminus\mathcal{B}.$$

Another intuitive property is that sensing quality is *monotonic* [1], i.e., $\mathrm{I}(\mathcal{A}) \leq \mathrm{I}(\mathcal{B})\ \forall\ \mathcal{A}\subseteq\mathcal{B}\subseteq\mathcal{V}$: The more locations we observe, the more information we will collect.

In [8], the authors exploited the submodularity and monotonicity property of MI to prove that if the discretization $\mathcal{V}$ is fine enough, and if the GP satisfies mild

[1]Monotonicity only approximately holds for mutual information [8]; however, this is sufficient for the scenario discussed in this paper.

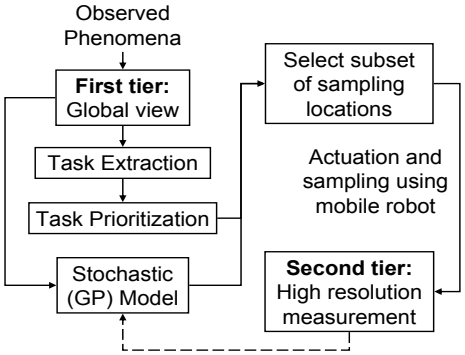


Fig. 1: An overview of a two-tier *MUST* architecture. A future data flow path is marked with a broken line.

Algorithm: *MUST*

Input: s, t, k, m

Output: A path $\mathcal{P} = s, \ldots, t$ over selected observation locations

begin

1 $\mathcal{T} \leftarrow tier1Tasks$;

2 $PT \leftarrow taskPrioritization(\mathcal{T})$;

3 $\Sigma \leftarrow learnGPmodel$;

for $1 \leq i \leq m$ **do**

4 $GS_i \leftarrow$ *GreedySubset* $(PT(i), \Sigma, k)$;

5 $\mathcal{P} \leftarrow tourOpt(GS)$;

return $\mathcal{P}$;

end

Algorithm 1: Efficient observation selection using *MUST*.

regularity conditions, greedily selecting locations is near-optimal. More specifically, the greedy algorithm (which we will call *GreedySubset*), after selecting the first i locations, $\mathcal{A}_i$, picks the location with maximum residual information, i.e., $v_{i+1} = \text{argmax}_v \, \text{I}_{\mathcal{A}_i}(\{v\})$ and sets $\mathcal{A}_{i+1} = \mathcal{A}_i \cup \{v_{i+1}\}$. Thus, *GreedySubset* iteratively adds locations yielding the most mutual information. The performance of the greedy algorithm for submodular functions was proven to have a near-optimal approximation guarantee using a result from [12].

III. Efficient Path Planning for Environmental Monitoring Applications

Next we discuss two path planning algorithms: 1) *MUST* - our proposed multiscale stochastic algorithm with greedy observation selection; and 2) the *OP heuristic* - a path planning algorithm known as the *Orienteering algorithm* proposed in [13].

A. MUST: Multiscale Stochastic Algorithm

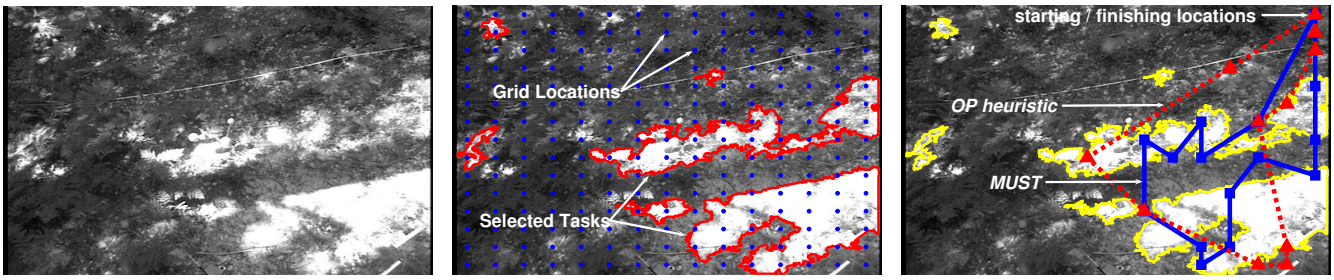
We now present the novel *MU*ltiscale approach with *ST*ochastic modeling, henceforth referred to as *MUST*. *MUST* extends the hierarchical multiscale algorithm proposed in [5] in two ways: a) a stochastic approach is used to model the observed environment; b) stochastic properties of the model are exploited to achieve near-optimal observation selection (i.e. within a region of interest) as proposed in [8].

First, we will briefly discuss the two-tier hierarchical algorithm proposed in [5]. We proposed exploiting different sensing modalities to observe a spatio-temporal dynamic environment with high fidelity. A low fidelity sensor with large spatial coverage and high temporal resolution is used to capture a "global" view of the environment under observation. When monitoring light intensity in a forest understory, a down-looking imager provides low fidelity data about light intensity distribution, i.e., the images identify the bright, sunlit regions. The fidelity of the information is low because the image provides a map of the reflected light, which is only a proxy for actual light intensity. The image, when collected frequently, can also provide high temporal resolution. The information from the first tier sensor is used to guide a mobile robot carrying a high fidelity sensor, which takes measurements in the regions of interest. When monitoring light intensity distribution, a PAR (Photosynthetically Active Radiation) sensor is guided to the sunlit regions to collect high fidelity measurements in these regions. Note that in [5] a uniform, raster scan path was performed to observe every location within the selected task, where as *MUST* observes only a subset of possible measurement locations within the task.

Algorithm 1 provides an outline for the *MUST* algorithm and Fig. (1) illustrates the corresponding schematic view. Similar to the proposed hierarchical approach in [5], the first tier sensor (an imager) provides regions of interest, referred to as tasks, for mobile robots to service (c.f. Line *1* of Algorithm 1). Task extraction is performed next. Gray scale images are first converted to binary (black and white) images via a threshold found using Otsu's method [14]. This threshold is chosen such that the variance is minimized among black and among white pixels. Then, edge detection is performed on these binary images by tracing the exterior edges of the white objects. Fig. (2b) illustrates this task extraction procedure on the image specified in Fig. (2a).

Next, we perform greedy task allocation, prioritizing the tasks based on the total area to observe (c.f. Line *2* of Algorithm 1). Tasks with an area smaller than an experimentally determined threshold are discarded as noise, i.e., not having large enough areas to constitute significant regions of interest. The key idea here is that observing larger tasks will result in observation locations that are both informative and relatively close together, which reduces traveling cost, thus providing a higher benefit to cost ratio. In addition to task allocation, information from the first tier sensor (pixel intensities for our example) is used to learn a stochastic Gaussian Process model (c.f. Section II). This is illustrated in Line *3* of Algorithm 1. The first tier sensor can also be used to periodically update the GP model; This is represented by a connection from first tier to the GP model in Fig. (1).

Pixel intensities do not directly measure the incident light responsible for photosynthesis, instead, they accurately recreate light intensity correlations in the observed environment. For a Gaussian Process model, only the correlations between observation locations are important. Eq. (2) and Eq. (3) show that mutual information can



(a) Image taken at 10:00 a.m. (b) Image with selected tasks (c) Illustrative paths for *MUST* and *OP heuristic*

Fig. 2: Illustration of path planning experiment using images collected from the San Jacinto Mountain Reserve. Grid locations in (b) represent the discrete observation locations used for path planning.

be calculated in a closed form by using the posterior covariance. Eq. (1) illustrates that the posterior covariance only depends on the correlations between the observation locations and not on the actual observation measurements. Therefore, our approach of learning the GP model from the first tier sensor information is highly efficient and accurate.

MUST takes as input a parameter m, which specifies the possible number of tasks to observe, and k, which specifies the number of locations to observe within each task. In the next step, for each task *GreedySubset* (c.f. Section II) uses the learned Gaussian Process model to greedily select k locations as per the mutual information criterion specified in Eq. (3) (c.f. Line *4* of Algorithm 1). In our example, we discretize the image into a 15x15 uniform grid of observation locations as illustrated in Fig. (2b). *MUST* greedily selects k locations from this uniform grid of observation locations. Note that Krause et al. [8] proved that greedy selection without a path cost constraint is near-optimal while optimizing a submodular, monotonic function. We then used the tour-opt heuristic from [15] to find the shortest path for the mobile robot (i.e. second tier that acts locally) to traverse, starting at a given location s, going through the selected observation locations $v_i; i \in m * k$, and finishing at a given location t (c.f. Line *5* of Algorithm 1). The execution time of *MUST* is polynomial in the number of observation locations.

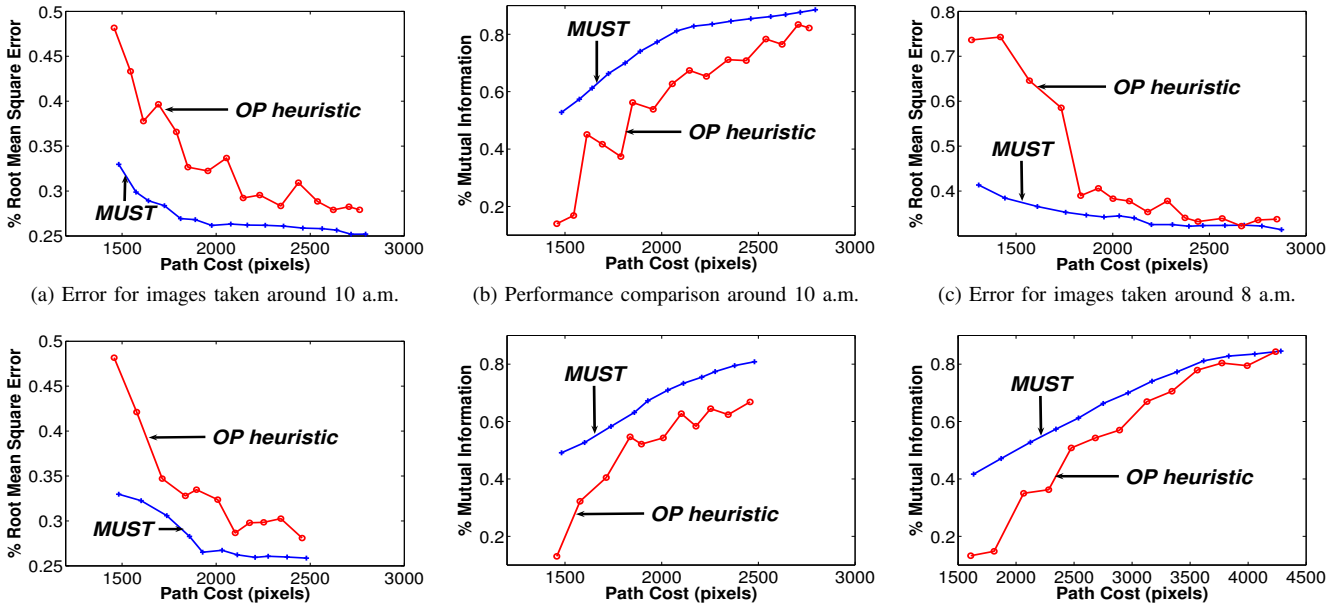
We associate with each observation location, v, in the selected grid, a *measurement cost* (where $C(v) > 0$), quantifying the expense, i.e, dwell time of obtaining a measurement at location v. When traveling between two locations, u and v, a robot incurs a *traveling cost* (where $C(u, v) > 0$), which we model using the euclidean distance criterion. The output path $\mathcal{P}$ from *MUST* is a sequence of $(l = m * k + 2)$ locations starting at node s, and finishing at node t. The cost $C(\mathcal{P})$ of path $\mathcal{P} = (s = v_1, v_2, \ldots, v_l = t)$ is the sum of measurement costs and traveling costs along the path, i.e., $C(\mathcal{P}) = \sum_{i=2}^{l-1} C(v_i) + \sum_{i=2}^{l} C(v_{i-1}, v_i)$. This path cost is then used as the input budget for comparison with the *OP heuristic*, which is discussed in Section IV.

It is important to note that *MUST* can be implemented using a variety of sensor combinations, task extraction and task prioritization methods. For light intensity monitoring, we have specified each of these approaches above. However, the first tier sensor need not be a single sensor; instead, it can be a set of uniformly distributed static sensors. A clustering approach could be used to extract regions of interest from the set of static sensors providing the "global" view. Additionally, task prioritization need not be greedy and based on area size. Our proposed stochastic modeling with greedy observation selection will provide similar performance guarantees with any of the task extraction and task allocation methods. We can also easily extend our proposed single robot *MUST* algorithm to the multiple robot setting using the *sequential-allocation* algorithm proposed in [7]. As proved in [7], *sequential-allocation* will also maintain the performance efficiency of our proposed algorithm for the multi robot-setting.

B. Orienteering Algorithm

We use the *orienteering algorithm* (hereafter referred to as the *OP heuristic*) discussed in [13] to efficiently compute efficient paths. This heuristic has been empirically found to be one of the best heuristics in a similar problem setting [16], and has furthermore been shown to perform effectively for monitoring light intensity [9]. For the sake of completeness, we briefly describe the *OP heuristic* algorithm for estimating efficient paths.

We start with an initial budget constraint that is in terms of the total feasible path cost. The *OP heuristic* algorithm works in two phases: initialization and improvement. In the initialization phase, an initial solution is calculated by constructing an ellipse over the entire set of observation locations with the given starting and finishing locations serving as the two foci of the ellipse. The available budget constraint is the length of the major axis. Now, only locations lying inside the ellipse satisfy the budget constraint requirement. Several paths are then explored over this set of observation locations, which also satisfy the budget constraint. Afterward, the paths are subjected to an exchange of locations between the possible paths, insertion/deletion of locations within a path, and finally, moving locations within the path. This exchange of observation locations seeks to find a path with an improved collected reward and a reduced path cost. In the improvement step, the total reward is allowed to decrease in search of a path with a larger total reward. Details of the *OP heuristic* algorithm are presented in [13].



(a) Error for images taken around 10 a.m.

(b) Performance comparison around 10 a.m.

(c) Error for images taken around 8 a.m.

(d) Error while *MUST* makes observations in the two highest priority tasks

(e) Performance comparison while *MUST* makes observations in the two highest priority tasks

(f) Comparison with measurement cost 3 times the distance between two observation locations

Fig. 3: Empirical evaluations comparing performance of *MUST* and *OP heuristic* for the light intensity application using images collected at around 10 a.m. (except in (c) when data is used from images collected around 8 a.m.). Path cost = measurement cost + pixels traversed, %RMS error = RMS error / (total RMS error when no observations are made), %MI = MI / (total MI when all observations are made in the observed tasks).

IV. Experiments and Analysis

In this section we validate the performance of *MUST* for the application of light intensity monitoring through both extensive simulations using field data and on an actual robotic platform, NIMS-PL (c.f. Section IV-A). We collected a series of images, every 17 seconds, throughout the day in a mixed conifer forest at the James San Jacinto Mountain Reserve in Southern California [2]. These images were collected using a down-looking imager capturing an area approximately 6 m in length by 4 m in width. We used a series of 11 images beginning at 10 a.m. as our representative dataset for the spatial and temporal variations occurring in the observed environment. For comparison, we also provide results for another set of 11 images collected beginning at 8 a.m.

For each experiment, we compared the performance of *MUST* and the *OP heuristic*. When comparing the root mean square error, we predicted the light intensity at the unobserved locations using Eq. (1) and compared the predicted intensity with the observed intensity. For the set of 11 images, we discretized each image into 15x15 uniformly spaced locations. We averaged the pixel intensities in the 5x5 neighborhood of each location to eliminate noise and emulate the observation made at the corresponding location. The observation locations are sufficiently far apart and, therefore, do not influence the correlations. Next, we empirically learned a Gaussian Process model, i.e., the corresponding covariance matrix Σ. We trained the GP model using 10 images from the set of 11 images. The 11^{th} image was used as the test image. Each image in the dataset was used as a test image with the remaining images serving as training images. At the end, the results were averaged - *leave-one-out cross-validation*.

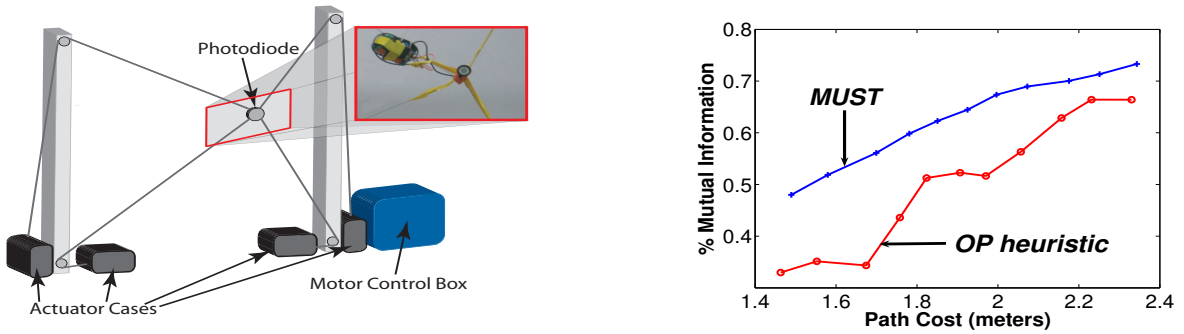
To compare the performance of *MUST* and the *OP heuristic*, we first calculated the path cost of the output path from *MUST* (as specified in Section III-A). This path cost was then given as the budget constraint to the *OP heuristic* algorithm and the corresponding heuristic path was evaluated for the total collected mutual information and the root mean square error. For simplicity, we set the starting location equal to the finishing location. We experimented with all four corners of the observed environment as the starting/finishing locations. We only present the results with the starting and finishing location specified in Fig. (2c) since the rest of the results exhibited similar patterns.

For each performance comparison, the x-axis represents the total path cost for the output path. Unless otherwise specified, we assumed the measurement cost to be the same as the traveling cost between two adjacent grid locations oriented along the x-axis. In the field, this is equivalent to a measurement cost of about two seconds, given the robot velocity = 0.25m/s and the distance between observation locations = 0.4m. This measurement cost allows multiple PAR measurements to be taken while the robotic node settles at the observation location. These PAR measurements are averaged, yielding an accurate measurement. For the simulation experiments, we present the path cost in terms of the number of pixels visited. For the experiments using NIMS-PL, the path cost is presented in meters.

A. Experiments in Simulation

Fig. (3) illustrates the results for the performance comparison of the *MUST* algorithm and the *OP heuristic*. We first

[2] http://www.jamesreserve.edu/



(a) Schematic of NIMS-PL system actuating an optical sensor

(b) Performance comparison using NIMS-PL

Fig. 4: Experiments using NIMS-PL robotic system with images collected around 10 a.m.

restricted *MUST* to make observations only in the highest priority task, i.e., $m = 1$ in Algorithm 1. Then, we varied the number of locations that could be observed within the highest priority task (i.e., variable k in Algorithm 1) from 5-20. For each of the output paths, $\mathcal{P}$, from the *MUST* algorithm, we calculated the total path cost and used that as the budget constraint for the *OP heuristic* to get its corresponding path with same path cost. Fig. (3a) and Fig. (3b) compare the root mean square error and the mutual information for *MUST* and the *OP heuristic*. As we increased the number of observed locations, the collected information asymptotically saturates (c.f. Fig. (3b)), which is in accordance with the submodularity property discussed in Section II. Similarly, in Fig. (3a) as the path cost increases, and therefore the number of observed locations increases, *MUST* asymptotically reaches the minimum achievable percent root mean square error. For Fig. (3a) and Fig. (3b), *MUST* outperforms the *OP heuristic* for the complete path cost range. To illustrate the general applicability of our comparative analysis, we present in Fig. (3c) the root mean square error comparison between the two algorithms for the set of 11 images collected beginning at 8 a.m.; *MUST* again outperforms the *OP heuristic*, which indeed was the general trend observed for experiments using other image datasets as input.

Next we increased the number of observed tasks, m, to 2 and performed similar experiments varying the number of observed locations within each task and comparing the corresponding performance of *MUST* and the *OP heuristic*. Fig. (3d) and Fig. (3e) compare the root mean square error and mutual information, respectively. *MUST* clearly provides better performance when compared with the *OP heuristic*. Another interesting observation from Fig. (3e) is that the collected information saturates at a higher path cost (approximately 2500 pixels) as compared with Fig. (3b) (observations in a single task) where the collected information saturates at a lower path cost (approximately 2000 pixels). Additionally, when comparing the same path cost across experiments, more information is collected when $m = 1$. This result supports our approach of prioritizing tasks based on their coverage area to increase the utility/cost ratio. Finally, we increased the measurement cost to three times the distance between two adjacent grid locations along the x-axis. Fig. (3f) compares the collected mutual information for both algorithms. In this scenario, as the path cost increases, both algorithms approach the asymptotic limit of maximum collected information.

For simplicity, these experiments analyze sampling performance over a relatively small area of 6 *m* in length by 4 *m* in width. *MUST* scales independently of the size of the area because of the first tier "global" view. Therefore, we would expect similar performance results for *MUST* when sampling much larger areas. On the other hand, the latency of adaptive sampling algorithms is prohibitive when observing phenomena that is dynamic in both space and time.

B. Real Robot Experiments

We validated our proposed *MUST* algorithm using the Networked Infomechanical System for Planar actuation (NIMS-PL) - a four cable-based robotic system used for actuation in planar workspaces [9]. NIMS-PL consists of four tension controlled cables actuating a sensor or sensor suite to provide planar spatial coverage. Planar operation is feasible both in a vertical plane as well as in a horizontal plane. The four cabled configuration enables quadrangular workspaces, resulting in a larger feasible workspace as compared with a three cabled configuration. Additionally, in aquatic deployments even larger spatial coverage is possible with NIMS-PL since the weight of the end-effector need not be supported by the cables but rather the water surface can support a buoyant sensor node. A schematic of the NIMS-PL system observing light intensity by actuating an optical sensor in a vertical plane is shown in Fig. (4a).

We now compare the performance of the two algorithms using the actual robotic system, NIMS-PL. To get the ground truth data, we projected each of the 11 images from the 10 a.m. dataset. A lab scale NIMS-PL system with a vertical workspace of $0.75m$ x $0.75m$ was used in the experiment. The end-effector of NIMS-PL was equipped with a photodiode to measure the incident light intensity from the projected images. Light intensity was sampled at a rate of $100Hz$ and measurements were relayed via Bluetooth. A continuous scan was performed at a velocity of $0.25m/s$ in the workspace. Measurements during the scan were averaged to get the corresponding data at each of the 15x15 grid locations uniformly distributed throughout the workspace.

We used the pixel intensity data from the projected images to learn the Gaussian Process model. Photodiode measurements made at the selected path locations were used to calculate the collected mutual information and to predict photodiode observations at unobserved locations for both *MUST* and the *OP heuristic*. Fig. (4b) compares the collected information for the two algorithms when performing path planning using NIMS-PL. *MUST* significantly outperforms the *OP heuristic* on a robotic system. This result is similar to the results obtained in simulation experiments, thus validating the simulation performance results.

V. Conclusions and Future Work

In this paper, we presented *MUST* - a multiscale stochastic algorithm with greedy observation selection. We explained the implementation of a two-tier *MUST* algorithm for monitoring light intensity to study plant growth. We used a stochastic Gaussian Process approach to model the uncertainty in the observed environment. We presented the prediction accuracy in terms of root mean square error as per the learned GP model. We then used the mutual information and root mean square error criteria to compare the performance of *MUST* with an efficient orienteering algorithm proposed in the literature. We performed extensive empirical evaluations using field datasets both in simulation and on a robotic system to validate the efficiency of our proposed algorithm. In the future, we plan to extend *MUST* for online path planning by using new observations to update the GP model.

VI. Acknowledgments

The authors would like to acknowledge Dr. Per Henrik Borgstrom for his assistance in operating NIMS-PL, which was used for the validation of simulation results by testing the *MUST* algorithm on a robotic system.

This material is based upon work supported in part by the US National Science Foundation (NSF) under Grants ECCS-0725441 and CNS-0331481. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] J. L. Hatfield, J. H. Prueger, and W. P. Kustas, "Spatial and temporal variation of energy and carbon fluxes in central iowa," *Agron J*, vol. 99, no. 1, pp. 285–296, 2007.

[2] E. Bauer and L. Keefer, "Spatial and temporal variations of nitrate concentration and load in an upper midwest agricultural watershed," *AGU Fall Meeting Abstracts*, pp. F1067+, Dec. 2008.

[3] R. A. Houghton, "Aboveground forest biomass and the global carbon balance," *Global Change Biology*, vol. 11, no. 6, pp. 945–958, 2005.

[4] M. Rahimi, R. Pon, W. Kaiser, G. Sukhatme, D. Estrin, and M. Srivastava, "Adaptive sampling for environmental robotics," in *ICRA*, 2004, pp. 3537–3544.

[5] A. Singh, D. Budzik, W. Chen, M. A. Batalin, M. Stealey, H. Borgstrom, and W. J. Kaiser, "Multiscale sensing: A new paradigm for actuated sensing of high frequency dynamic phenomena," in *IROS*, 2006, pp. 328–335.

[6] A. Singh, R. Nowak, and P. Ramanathan, "Active learning for adaptive mobile sensing networks," in *IPSN*, 2006, pp. 60–68.

[7] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, "Efficient planning of informative paths for multiple robots," in *IJCAI*, 2007, pp. 2204–2211.

[8] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *JMLR*, vol. 9, pp. 235–284, 2008.

[9] P. H. Borgstrom, A. Singh, B. L. Jordan, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, "Energy based path planning for a novel cabled robotic system," in *IROS*, 2008, pp. 1745–1751.

[10] C. E. Rasmussen and C. K. Williams, *Gaussian Process for Machine Learning*, ser. Adaptive Computation and Machine Learning. MIT Press, 2006.

[11] W. Caselton and J. Zidek, "Optimal monitoring network design," *Statistics and Probability Letters*, 1984.

[12] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of the approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.

[13] I. Chao, B. Golden, and E. Wasil, "A fast and effective heuristic for the orienteering problem," *EJOR*, vol. 88, pp. 475–489, 1996.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE TSMC*, vol. 9, no. 1, pp. 62–66, 1979.

[15] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, pp. 2245–2269, 1965.

[16] Y.-C. Liang, S. Kulturel-Konak, and A. E. Smith, "Meta heuristics for the orienteering problem," in *CEC*, 2002, pp. 384–389.