Fast Marching Adaptive Sampling

Nicholas R. J. Lawrance, Jen Jen Chung, and Geoffrey A. Hollinger

***Abstract*—A challenging problem for autonomous exploration is estimating the utility of future samples. In this paper, we consider the problem of placing observations over an initially unknown continuous cost field to find the least-cost path from a fixed start to a fixed goal position. We propose the adaptive sequential sampling algorithm FMEx to successively select observation locations that maximize the probability of improving the best path. FMEx evaluates a set of proposed observation locations using a novel fast marching update method and selects a location based on the probabilistic likelihood of improving the current best path. Simulated results show that FMEx finds lower-cost paths with fewer samples than random, maximum variance and confidence bound sampling. We also show results for sampling bathymetric data to find the best route for a submarine cable. In problems where sampling is expensive, FMEx selects observation locations that minimize the true path cost.**

***Index Terms*—Motion and path planning, probability and statistical methods, reactive and sensor-based planning.**

I. Introduction

ROBOTS often require models of their operating environments to make reasoned mission decisions. In unknown environments, the world is often sampled in order to generate predicted models, but in many cases sampling comes at a cost. This sampling cost must be traded off against the cost of calculating and executing a suboptimal mission plan based on an incorrect model. Consider sampling the bathymetry of an ocean channel to find the best path along which to lay a cable [1], where each bathymetric sample might require long deployments of an underwater vehicle. Alternatively, consider planning a safe evacuation route through a wildfire [2] where sampling the local fire intensity requires aerial deployment of a single-use sensor [3]. In such circumstances, incorporating knowledge of the planning goals into the sampling strategy can allow for more targeted and effective sampling of the environment with limited observations.

Previous information gathering sensor placement problems have primarily focused on selecting samples to maximize information (or reduce uncertainty) about the field being studied, or in finding particular values in the field. These metrics for driving exploration have been shown to improve the model accuracy over the entire space. However, in this paper we consider the problem of sampling an (initially unknown) cost field to find the least-cost path between two points. Thus there is an additional layer of abstraction, since although some observations may be informative about the field, they may also be unlikely to change the path cost, and hence are relatively poor use of a limited sampling budget.

This paper describes a novel, efficient path cost update method paired with a Gaussian Process (GP) regression model to sequentially select samples that maximize the likelihood of reducing path cost. The proposed planning and update technique, FMEx, uses Fast Marching (FM) [4] to approximate the cost-to-come function of continuous spaces with a discrete grid. FMEx is based on two novel contributions to solve the minimum-cost replanning problem. First, we propose biFM, an adaptive FM search that efficiently computes the least-cost path based on local map updates. Second, we present an allocation algorithm that sequentially selects observations with the goal of minimizing the expected path cost.

Experiments on simulated cost fields demonstrate that FMEx finds lower-cost paths with fewer samples when compared to random, maximum variance and confidence bound-based sampling techniques. biFM is also shown to reduce mean replanning time by 77.9% compared to existing FM-based dynamic replanning graph search techniques.

II. Related Work

This work is at the intersection of three major areas of robot planning research: adaptive sampling, FM path search and dynamic replanning. We draw ideas from informative exploration to develop a mission-targeted sampling strategy, and we leverage computationally efficient search properties from FM to perform efficient replanning.

A. Sampling for Exploration

Exploration sampling problems generally involve attempting to model an unknown or partially known phenomenon using a limited budget of discrete samples. There are challenges in both the modeling and sampling strategies. Probabilistic models such as GPs are commonly used due to their ability to provide Bayesian statistics on the model quality (uncertainty) as well as a current best estimate. Model quality estimates can also be used to define sampling strategies that attempt to minimize model error [5]. In information gathering problems relatively simple strategies such as sequential greedy observation selection can perform near-optimally [6].

Manuscript received September 1, 2016; accepted December 28, 2016. Date of publication January 10, 2017; date of current version February 2, 2017. This paper was recommended for publication by Associate Editor S. Chakravorty and Editor N. Amato upon evaluation of the reviewers' comments. This work was supported in part by ONR under Grant N00014-14-1-0509 and in part by NASA under Grant NNX14AI10G.

The authors are with the Robotics Program, School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331 USA (e-mail: nicholas.lawrance@oregonstate.edu; jenjen.chung@oregonstate.edu; geoff.hollinger@oregonstate.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2017.2651148

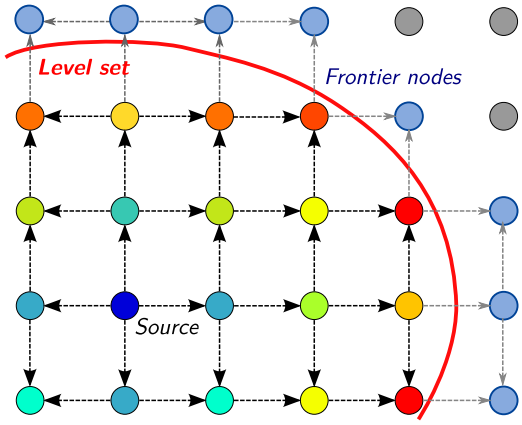

Fig. 1. FM search from a single source node. Node colors indicate the time of arrival or cost-to-come of accepted nodes, black arrows indicate inheritance relationships. The grey arrows and pale blue nodes indicate frontier nodes that have had their costs updated but not yet accepted. Nodes that are descendants of a particular node are termed 'downwind'.

Particularly relevant work focuses on estimating the *value* of a sample, in terms of the quality of the model or a mission-specific metric. Previous work has explored mission criteria such as maintaining communication [7] and improving model quality for reinforcement learning [8] in motion planning. Our work differs from prior work because the only utility is path cost, and we leverage information to improve the likelihood of reducing path cost. Previous work [9] has examined selection strategies based on Bayesian optimization, where sampling based on the Upper Confidence Bound (UCB), the linear sum of mean and uncertainty, is used to find the highest value using a GP. Such techniques are not necessarily well-aligned for finding low-cost paths as they focus on model uncertainty, not the effect of the model on path cost.

B. Fast Marching

The FM method was proposed in [4] as a solution to estimating level-set problems. This type of problem assumes a continuous scalar function $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^+$ that represents the isotropic 'inverse speed' that a wavefront passing through point $\mathbf{x} \in \mathbb{R}^n$ would travel at. The set of points that are the same minimum time away from a start location are known as a level set (a circle for constant f in $\mathbb{R}^2$). However, solving the continuous problem analytically is difficult, and FM provides an approximation to finding level sets using a discrete grid and approximate solution to the eikonal equation. The result is similar to common graph-based planners such as Dijkstra and A* where nodes are accepted in the order of first possible arrival using a priority queue. However, in FM the edge costs are calculated using the sampled f values from multiple neighboring cells.

A typical FM search is shown in Fig. 1. At the end of an FM search, the first time-of-arrival is stored at each node, and the whole field represents a discrete approximation to the continuous cost-to-go function from the start nodes. The lowest-cost path is recovered by interpolating the discrete estimate to recover cost gradients and following the steepest gradient downwards from a goal.

It can be readily seen that the problem of finding the lowest-cost path for a mobile robot in an obstacle field with variable traversal costs can be considered as a level-set problem. Sethian [4] showed that FM retains the $\mathcal{O}(n \log n)$ (in the number of cells n) computational complexity of Dijkstra's method, but generates a smooth approximation to the continuous field. Further work [10] showed that similar methods, known as ordered upwind (OU), can also be applied to anisotropic fields (where traversal cost is direction dependent). Since then, many works in robotics literature have used FM and OU methods to solve robotic planning problems with continuous cost spaces [11]–[14].

C. Dynamic Replanning

Our work follows from methods developed for fast graph replanning, including Field-D* [15] and E* [16]. Both Field-D* and E* were developed as discrete grid planners that approximate a continuous space rather than being restricted to fixed graph edge directions. Field-D* uses linear interpolation to estimate the cost at arbitrary positions on cell edges. E* uses the FM update to produce a discrete estimate of the continuous cost-to-come.

Field-D* and E* use an inheritance relationship to update the cost calculations of affected nodes when a change is observed. In Field-D*, backpointers are used to record a parent node, and in E* the concept is expanded to an inheritance graph because nodes can have multiple parents. Only nodes downwind (see Fig. 1) of a cost change are affected. However, using the inheritance graph to find the nodes affected by a cost update can be nearly as costly as the original search. Further, the number of nodes expanded is dependent on where in the map the update occurs. Nodes with lower costs (closer to a start node) generally have more downwind dependent nodes. We mitigate this issue using the bi-directional search described in Section IV-E.

III. Problem Statement

Consider a continuous region $\chi \in \mathbb{R}^n$ with an associated isotropic positive scalar cost field $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^+, \forall \mathbf{x} \in \chi$, as in the FM description in Section II-B. We consider the problem of finding the lowest-cost path $\mathcal{P}^*$, through χ from start location $\mathbf{x}_s$ to goal $\mathbf{x}_g$. The cost of a continuous curve (path) $\mathcal{P}$ parameterized by s along its length, can be written as the line integral of f along $\mathcal{P}$, such that

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \int_{\mathcal{P}} f ds. \tag{1}$$

This form applies to many continuous planning problems where the goal is to find a minimum-cost path. From earlier discussions, we know that FM can be used to approximate $\mathcal{P}^*$ using a grid-based approximation of f.

In this work, we consider the problem of sequentially sampling f to find (the best approximation of) $\mathcal{P}^*$. We assume that the problem starts with an estimate of the mean of the field $\bar{f}$, but no observations of the true field. This is a sensor placement problem in that all $\mathbf{x} \in \chi$ are assumed to be equally costly to visit. Observations are made by selecting a location $\mathbf{x}_i$ and sampling from the true cost field using the observation function, $z_i = z(\mathbf{x}_i)$. We assume a normally distributed observation model with fixed variance, $z(\mathbf{x}) \sim \mathcal{N}(f(\mathbf{x}), \sigma_n^2)$. The set of observations

Algorithm 1: Shortest path exploration with fast marching.

1: **procedure** FMEx
2: **while** Remaining observation budget > 0 **do**
3: Estimate cost map from GP
4: Select p locations from the cost field
5: At each location, draw q samples from GP and use BIFM to estimate expected path cost
6: Make observation at best location and update GP
7: Estimate cost map using GP and extract path

up to time t is denoted $Z_t = \{(\mathbf{x}_i, z_i) \mid i = 1, \dots t)\}$, and $Z_0 = \emptyset$. The goal is to use a limited sampling budget to calculate a path $\mathcal{P}'$ that minimizes the true path cost.

Further, assume we have access to a function $\mathcal{C}(Z)$ that estimates the lowest path cost conditioned on a set of observations Z. Then, at each time t, the goal is to find the location $\mathbf{x}^*_{t+1}$ that is most likely to find the lowest-cost path. By performing the expectation over possible observations z, the objective can be written

$$\mathbf{x}^*_{t+1} = \arg\min_{\mathbf{x}} \; \mathbb{E}_z \left[\mathcal{C} \left(Z_t \cup \{(\mathbf{x}, z)\} \right) \right]. \tag{2}$$

The proposed FMEx method uses a combination of GP estimation and a novel adaptive replanning method based on FM to sample locations and observations from the field, estimate the expectation in (2) and select the location that minimizes the expected path cost.

IV. FMEx Method

We structure our proposed algorithm, FMEx, as a sequential process with four main steps (Algorithm 1). First, a grid-based estimate of the cost map is generated using a Gaussian Process (GP) conditioned on the current set of observations. Second, a set of potential observation locations are selected from the field. Third, the expectation of the path cost in (2) is estimated using the proposed biFM algorithm by sampling from the GP at each location and finding the lowest path cost using FM. The location with the lowest expected path cost is selected, the robot makes a (noisy) observation of the true cost at that location, adds the sample to the GP, and repeats. Once the observation budget is exhausted, the best path is extracted from the final cost field estimate.

A. Cost Map Estimation

We propose using GP regression to approximate $f(\mathbf{x})$:

$$f(\mathbf{x}) \sim \mathcal{GP}(\bar{f}, k(\mathbf{x}, \mathbf{x}')) \tag{3}$$

where $\bar{f}$ is a known constant (the estimated mean field cost), and we use the squared exponential covariance for k, a stationary radial basis function. During execution, the model is conditioned on the current observation set Z_t. A GP was selected because it generates smooth approximations of the cost field, handles noisy observations and provides variance estimates. FMEx uses the GP model conditioned on Z_t to construct a posterior estimate of the cost map $\hat{f}_{Z_t}$ by sampling the mean and variance from the GP at each grid point. A detailed discussion of GP regression is beyond the scope of this paper and interested readers are directed to [17].

B. Observation Locations

FMEx uses a sampling approach to approximate the expectation in (2) given an observation location $\mathbf{x}$. To generate a set X_p of possible locations, we use uniform random sampling to select a fixed number (p) of locations from χ at each iteration of the main FMEx loop. For computational efficiency, these locations are currently restricted to the set of (non-obstacle) grid points used to generate the GP cost map in the previous step. However, the method could use any set of possible observation locations and we will consider more directed sampling approaches in future work.

C. Estimating Expected Path Cost Using biFM

Using the GP to estimate the cost map and then using FM to find the minimum path cost can effectively be treated as a function $\mathcal{C}(Z)$ that estimates the best path cost given a set of observations. However, to select a location for making a future observation, we would like to reason over the possible *future* path costs given a target observation location $\mathbf{x}'$. We can do this by calculating the expectation of the path cost $\mathbb{E}_z \left[\mathcal{C} \left(Z_t \cup \{(\mathbf{x}', z)\} \right) \right]$, over the possible observations z,

$$\mathbb{E}_z \left[\mathcal{C} \left(Z_t \cup \{(\mathbf{x}', z)\} \right) \right] = \int \mathcal{C} \left(Z_t \cup \{(\mathbf{x}', z)\} \right) p(z) dz. \tag{4}$$

This equation poses two major challenges. First, we require the observation likelihood $p(z)$ for a target observation location. Fortunately, the GP provides this distribution in the form of the predictive mean and variance estimates. The second challenge is solving the integration over the cost function $\mathcal{C}$. We do not have an analytically tractable $\mathcal{C}$ function, but using the GP cost map estimate and FM we can estimate $\mathcal{C}(Z)$ for a given set of observations. This suggests a sampling-based approximation by discretely sampling from the $p(z)$ distribution produced by the GP,

$$\mathbb{E}(C_{\mathbf{x}'}) \approx \frac{\sum_{i=1}^{q} \mathcal{C} \left(Z_t \cup \{(\mathbf{x}', z)\} \right) p(z_i)}{\sum_{i=1}^{q} p(z_i)}. \tag{5}$$

Section V demonstrates two proposed methods for sampling from $p(z)$: a Monte Carlo sampler which directly samples a normal distribution and a sampler using fixed standard deviation increments above and below the mean. Then, for each $\mathbf{x}_p$ in the set of proposed observation locations X_p, we draw q samples from $p_{\mathbf{x}_p}(z)$, the GP posterior at $\mathbf{x}_p$, using the chosen sampler. This yields a set of proposed observations at $\mathbf{x}_p$ and their associated probabilities, $Z_{\mathbf{x}_p} = \left\{ (z_j, p_{\mathbf{x}_p}(z_j)) \mid j = 1, \dots q \right\}$. Then, the full set of proposed observations for all locations can be written,

$$Z_p = \left\{ (\mathbf{x}_p, Z_{\mathbf{x}_p}) \; \forall \; \mathbf{x}_p \in X_p \right\}. \tag{6}$$

Equation (5) could be evaluated naïvely by incorporating each observation in Z_p into a separate GP model, sampling the GP over the grid to generate the cost map and finally using FM to find the new minimum path cost. However, this would be computationally expensive, and would result in redundant computation because in many cases the best path cost would change only a little or not at all because the observation had limited

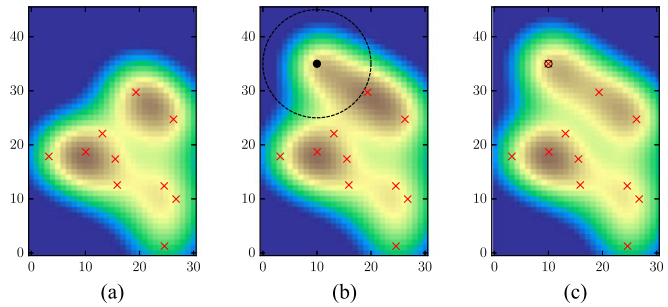

Fig. 2. The local cost update using the D polynomial estimate (7) only changes costs in the interior of the circle in 2 b. This allows biFM to update a much smaller set of costs than would be required by the full GP update. (a) GP field estimate with 10 samples ($\times$). (b) Field estimate using local cost update D (8). (c) Field estimate for full GP update.

affect on the costs near the current best path. To mitigate this excess computation, we propose a solution that uses a novel FM repair technique called biFM. biFM consists of two major components: a localized map update and an FM cost-to-come repair technique that reuses areas of an existing valid search to speed up cost calculations when there are local changes made to a cost map.

D. Localized Cost Map Update

To avoid updating the GP over the whole cost map for each proposed observation, we model the map update as a local cost modification centered on the observation location. Consider a proposed observation $(\mathbf{x}_p, z_p)$. We use a second-order polynomial approximation of the squared exponential covariance function from [17],

$$D(\mathbf{x}) = \begin{cases} (1-r)^4 (4r+1) & \text{if } r < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $r = \|\mathbf{x} - \mathbf{x}_p\|/R$, the distance between $\mathbf{x}$ and $\mathbf{x}_p$ normalized by a distance scale R. In our work we fix R to match the length scale from the GP covariance function. The D function has compact support, so that the cost is only varied up to distance R from $\mathbf{x}_p$. By scaling D by the new observation magnitude and summing with the existing cost field estimate from the GP ($\hat{f}_{Z_t}(\mathbf{x})$), we create an updated cost field $\hat{f}_p(\mathbf{x})$ for each proposed observation $(\mathbf{x}_p, z_p)$,

$$\hat{f}_p(\mathbf{x}) = \hat{f}_{Z_t}(\mathbf{x}) + \left(z_p - \hat{f}_{Z_t}(\mathbf{x})\right) D(\mathbf{x}). \tag{8}$$

This modifies a compact region to reflect the proposed observation and leaves a large proportion of the field unchanged. Fig. 2 shows the difference between the estimated field $\hat{f}_p$ using D (Fig. 2(b)) and the resulting map if the same observation were added to the GP model (Fig. 2(c)). The local cost update approximates a full GP update but can be computed in a shorter and constant time. In our experiments using the $\hat{f}_p$ approximation from D took less than 1% of the time required for the full GP update, allowing many more proposed observations to be tested. As shown in the next section, the best path cost C' over the updated cost field $\hat{f}_p$ can be efficiently computed if $\hat{f}_{Z_t}$ has already been searched.

Algorithm 2: biFM - Given a graph G, start and goal nodes $\mathbf{x}_s$ and $\mathbf{x}_g$, cost function g and proposed observations Z_p.

1: procedure biFM $(G, \mathbf{x}_s, \mathbf{x}_g, g, Z_p)$
2: $A_s, I_s \leftarrow$ FM $(G, g, \mathbf{x}_s)$ ▷ Forward search
3: $A_g, I_g \leftarrow$ FM $(G, g, \mathbf{x}_g)$ ▷ Backward search
4: $C_F \leftarrow A_s(\mathbf{x}_g)$ ▷ Current lowest path cost
5: **for all** $\mathbf{x}_p \in Z_p$ **do** ▷ For each obs. location
6: $C_{\mathbf{x}_p} \leftarrow 0$, $P_{\mathbf{x}_p} \leftarrow 0$
7: **for all** $(z, p(z)) \in Z_p$ at location $\mathbf{x}_p$ **do**
8: $\Delta \leftarrow (z - g(\mathbf{x}_p))$
9: $\mathcal{D} \leftarrow \{(\mathbf{x}, \Delta D(\mathbf{x}))\ \forall \mathbf{x} \mid r(\mathbf{x}, \mathbf{x}_p) < 1\}$
10: $C_z \leftarrow$ UPDATE $(G, A_s, I_s, A_g, I_g, g, \mathcal{D}, C_F)$
11: $C_{\mathbf{x}_p} \leftarrow C_{\mathbf{x}_p} + p(z) C_z$ ▷ Prob. path cost sum
12: $P_{\mathbf{x}_p} \leftarrow P_{\mathbf{x}_p} + p(z)$ ▷ Likelihood sum
13: $\mathbb{E}(C_{\mathbf{x}_p}) \leftarrow C_{\mathbf{x}_p} / P_{\mathbf{x}_p}$ ▷ Expected path cost (5)
14: $\mathbf{x}_p^* \leftarrow \arg\min_{\mathbf{x}_p} \mathbb{E}(C_{\mathbf{x}_p})$
15: **return** $\mathbf{x}_p^*$

E. Bi-directional Fast Marching for Local Map Updates

A local map update can be considered as a replanning problem, where large parts of an existing cost calculation remain valid but some areas are invalidated by the cost change. Standard FM is designed for single-query searches and is not well-suited to replanning. A naïve implementation for finding the best path cost after a cost update would require restarting the search from scratch. This is potentially expensive for large grids. We propose biFM for computing the best path cost on locally-updated maps.

1) biFM (Algorithm 2): exploits the inheritance feature of FM using a two-stage process. Firstly (lines 2–5), the algorithm performs a complete FM search over the supplied cost function from both the start and goal nodes. The FM$G, g, \mathbf{x}_s$ function is a standard FM search over graph G, cost function $g = g(\mathbf{x})$ and start cell(s) x_s that returns the cost-to-come $A(\mathbf{x})$ and inheritance list $I(\mathbf{x})$ for each $\mathbf{x} \in G$. Thus, biFM calculates the lowest cost from the start to every other cell, and from the goal to every other cell. During each iteration of FMEx, biFM is called using the cost function $g = \hat{f}_{Z_t}$ from the GP predicted mean conditioned on observations Z_t.

Next, for each proposed observation we generate the set $\mathcal{D}$ of locations where the cost map has been updated, and their corresponding values using (7) (line 9). Then, UPDATE calculates the new best path cost C_z, which may remain unchanged. The path costs of all proposed observations at each $\mathbf{x}_p$ are used to estimate the expected path cost (line 13) as in (5), and the location with the lowest expected path cost is returned. Note that the jobs invoked by the loops at lines 6 and 8 are independent and could be computed in parallel.

2) Update (Algorithm 3): is used to calculate the change in best path cost associated with a map update $\mathcal{D}$. UPDATE requires as input: the graph G, the FM costs A_s & A_g, inheritance relationships I_s & I_g, the original cost map g, updated cost set $\mathcal{D}$ and the current best path cost C_F. To find the updated best path cost, UPDATE determines which cells had costs modified, and then starts a new FM search over that region. However, this function limits the search by calculating bounds on the change

Algorithm 3: Bi-directional fast march update.

1: **function** Update $(G, A_s, I_s, A_g, I_g, F, \mathcal{D}, C_F)$
2: $Q \leftarrow \emptyset$ ▷ Frontier priority queue
3: $C_{\min,s} \leftarrow \min\{A_s[\mathbf{x}] : \forall \mathbf{x} \in \mathcal{D}\}$ ▷ Min cost to $\mathbf{x}_s$
4: $C_{\min,g} \leftarrow \min\{A_g[\mathbf{x}] : \forall \mathbf{x} \in \mathcal{D}\}$ ▷ Min cost to $\mathbf{x}_g$
5: **if** $C_{\min,s} + C_{\min,g} > C_F$ **then**
6: **return** C_F ▷ Return if > than existing best
7: **if** $C_{\min,s} > C_{\min,g}$ **then** ▷ If update closer to $\mathbf{x}_g$
8: $A' \leftarrow A_s, C_A \leftarrow C_{\min,s}$ ▷ Search forward
9: $B' \leftarrow A_g, C_B \leftarrow C_{\min,g}$
10: **else** ▷ Update closer to x_s
11: $A' \leftarrow A_g, C_A \leftarrow C_{\min,g}$ ▷ Search backward
12: $B' \leftarrow A_s, C_B \leftarrow C_{\min,s}$
13: **if** $\mathcal{D}[\mathbf{x}] > 0\ \forall \mathbf{x} \in \mathcal{D}$ **then** ▷ Update increased costs
14: $M, A' \leftarrow$ InvalDownwind $(\mathcal{D}, I_s, A')$
15: **else**
16: $M \leftarrow \emptyset$
17: **for all** $\mathbf{x}_D \in \mathcal{D}$ **do**
18: $M \leftarrow M \cup \{\mathbf{x} : \forall \mathbf{x} \in I_s.parents[\mathbf{x}_D] \setminus \mathcal{D}\}$
19: **for all** $\mathbf{x}_m \in M$ **do** ▷ Add interface nodes to queue
20: $Q.push(\mathbf{x}_M : A'[\mathbf{x}_M])$
21: $C' \leftarrow \min\{A'[\mathbf{x}] + B'[\mathbf{x}] : \forall \mathbf{x} \in A'\}$
22: $C_l \leftarrow C' - C_B$ ▷ Cost bounds
23: $C' \leftarrow$ Research $(G, Q, A', B', F, \mathcal{D}, C_l, C_B)$
24: **return** C'

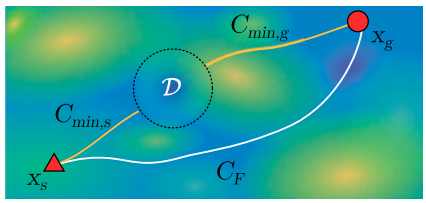

Fig. 3. Update diagram showing the bounding costs for an update $\mathcal{D}$ to an existing map. The previous best path over the original map has cost C_F. The lowest valid costs from the changed region to the start ($C_{\min,s}$) and goal ($C_{\min,g}$) can be used to bound or reject the subsequent search if the updated map could not contain a lower-cost path ($C_{\min,s} + C_{\min,g} > C_F$).

in path cost. Lines 3, 4 of Algorithm 3 calculate the minimum cost-to-come from the edge of the changed region to both the start and the goal, $C_{\min,s}$ and $C_{\min,g}$ respectively, as shown in Fig. 3). These node costs remain valid because they are upwind of the cost change. They also bound the lowest possible cost for a path passing through the updated region by $C_{\min,s} + C_{\min,g}$ (because the cost of traversing the updated region is ≥ 0). If this value is larger than the existing bound C_F then the original cost (if it remains valid) is retained (line 6) and the algorithm immediately returns. This often happens for updates that are distant from the best path, where the cost of traveling to and from the update region is greater than the current best path.

Next, we use FM to update node costs that were changed by the cost update. Since existing searches in both directions are available, we select the search direction which invalidates the fewest downwind nodes (line 7). If the cost change is an increase ($\Delta > 0$) then the InvalDownwind function invalidates any cost estimates in A' that are downwind of the updated nodes using the inheritance graph I_s. The downwind cost invalidation can be seen in the video associated with this paper. InvalDownwind also returns a set of *interface* cells M, that are parents of invalidated nodes and that are not in $\mathcal{D}$. These are valid costs in the forward search that can be pushed onto the search queue (line 20). If the cost update was a decrease ($\Delta < 0$), then all existing node costs remain valid because they will be overwritten during the FM Research by a lower-cost front. In this case, M only contains parents of nodes in $\mathcal{D}$ that are not themselves in $\mathcal{D}$ (line 18).

Algorithm 4: Local fast march update.

1: **function** Research $(G, Q, A, B, F, \mathcal{D}, C_l, C_B)$
2: **while** $\neg Q.empty()$ **do**
3: $(\mathbf{x}, c_\mathbf{x}) \leftarrow Q.pop()$
4: **if** $c_\mathbf{x} > C_l$ **then**
5: **return** $C_l + C_B$ ▷ Over budget
6: **else if** $B[\mathbf{x}] <= C_B$ **then**
7: **return** $c_\mathbf{x} + B[\mathbf{x}]$ ▷ Better path found
8: $A[\mathbf{x}] \leftarrow c_\mathbf{x}$
9: **for all** $\mathbf{y} \in G.neighbors(\mathbf{x})$ **do**
10: $\tau \leftarrow F[\mathbf{x}] + \mathcal{D}[\mathbf{x}]$ ▷ Transition cost
11: $c_\mathbf{w} \leftarrow c_\mathbf{x} + \tau + \delta$ ▷ Any $\delta > 0$
12: **for all** $\mathbf{w} \in G.neighbors(\mathbf{y}) \cap A$ **do**
13: **if** $(\mathbf{y}, \mathbf{w}) \perp (\mathbf{y}, \mathbf{x})$ **then**
14: $c_\mathbf{w} \leftarrow A[\mathbf{w}]$
15: **if** $\tau > |c_\mathbf{x} - c_\mathbf{w}|$ **then**
16: $c_\mathbf{y} = \frac{1}{2}\left(c_\mathbf{x} + c_\mathbf{w} + \sqrt{2\tau^2 - (c_\mathbf{x} - c_\mathbf{w})^2}\right)$
17: **else**
18: **if** $c_\mathbf{x} <= c_\mathbf{w}$ **then**
19: $c_\mathbf{y} \leftarrow c_\mathbf{x} + \tau$
20: **else**
21: $c_\mathbf{y} \leftarrow c_\mathbf{w} + \tau$
22: **if** $\mathbf{y} \notin A$ or $A[\mathbf{y}] > c_\mathbf{y}$ **then**
23: $Q.push(\mathbf{y} : c_\mathbf{y})$ ▷ Lower-cost update

The interface list is used to populate the frontier queue Q to continue the FM search from the remaining valid nodes. The bound C' (line 21) is the current lowest valid path cost. C_l is a local bound on how much the new search can expand before it would breach the C' bound. The frontier list Q, accepted nodes A' and bounds C' and C_l are then used to run a partial FM search using the Research function.

3) ReSearch (Algorithm 4): is a basic FM search that continues the search from nodes in Q to determine if a new best path cost could lie in the updated map. The only difference from a normal FM search is that the search can be terminated early if either the cost increases above the existing minimum C' (line 4), or by reaching a valid existing node from the other direction (line 6).

Fig. 4 shows the search time and total number of node expansions for FM, E*, and biFM for a set of 1000 randomized updates on the same map (the 100×100 map shown in Fig. 5). Each update was at a random location in the map and used the polynomial cost estimate update (7) with radius 14. Total

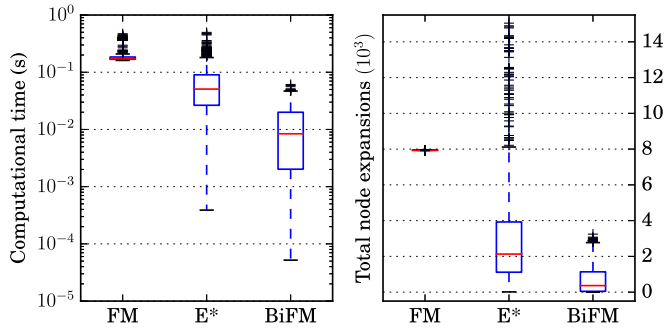

Fig. 4. Total node expansions and computational time for randomized map updates. Red lines indicate the median, blue boxes indicate data quartiles.

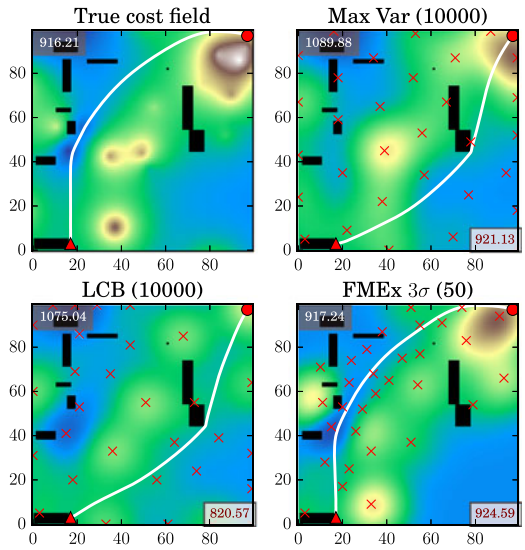

Fig. 5. Cost field and path estimates after $t = 30$ samples. The true field and best path are shown top left. Previous observations are shown as $\times$, and the current field estimate is the blue-green-brown field (low to high cost, respectively). The current best path estimate is shown for each method, along with associated true (top left, white text) and estimated (lower right, red text) path costs.

node expansions includes nodes expanded during the downwind search. biFM provides a reduction in the mean number of node expansions of 91.4% compared to standard FM and 77.9% compared to E*. Note that biFM has a higher initial search cost (twice that of FM), but offers strong advantages when there are more than two updates (such as in the FMEx algorithm). All methods return near-identical cost maps. E* sometimes takes longer than FM because if the update occurs near the goal the entire map is removed in the downwind sweep and must be recalculated.

F. Computational Complexity

We consider the computational complexity of the FMEx algorithm for selecting a single observation location (the contents of the while loop in Algorithm 1). Firstly, the GP update is an $\mathcal{O}(t^3)$ operation in the number of previous observations t. Next, a complete FM search is performed on the GP-estimated cost map, which is $\mathcal{O}(n \log n)$ in the number of cells in the grid n. Then, for each of p observation locations FMEx calculates a discrete estimate of the expectation in 5 with q observation samples at each location. For each sample, biFM calculates the updated path cost, an $\mathcal{O}(m \log m)$ operation in the number of cells updated (m varies depending on the update size but is commonly at least an order of magnitude less than n, as shown in Fig. 4). The total computational complexity per iteration is $\mathcal{O}(t^3 + n \log n + pqm \log m)$. In most cases, n, m, p and q remain constant, and t increases as observations are added. Generally, the third component dominates the complexity, but the GP term could dominate for large t. Thus the method scales approximately linearly with p and q which are user-defined. There is an interesting relationship to decide the values of p and q, as increasing p allows FMEx to select from more locations in the field, and increasing q provides better approximations of the path cost expectation at each location. Some discussion of this is provided in the results, but we have not yet developed a formal method for selecting these values given a problem instance.

V. Results

To show that FMEx generates a sampling strategy that minimizes path cost given an observation budget, we compared against three existing methods for selecting sample locations from a GP. All comparison methods used a GP with the same covariance function, hyperparameters and mean. Hyperparameters were selected at the start of the problem and were not retrained during exploration. We compare against random sampling, maximum variance sampling and lower confidence bound (LCB) sampling. Maximum variance sampling is a common strategy for minimizing the GP prediction error [18]. However, this has the effect of tending to sample the area uniformly for a radial basis covariance function. LCB sampling is used to find low values of a field, which helps identify low-cost regions but tends to ignore high-cost regions that strongly affect path cost.

We show results for three variants of FMEx with different sampling schemes to sample from the (normal) GP posterior at a target location $\mathbf{x}_p$ for approximating the discrete expectation in (5). FMEx1σ samples at one standard deviation above and below the mean ($q = 2$), and FMEx3σ samples at $\pm 1, 2$ and 3σ around the mean ($q = 6$). The FMExMC6 sampler draws $q = 6$ samples from a normal distribution (the GP posterior). Generally, we would expect performance to improve with increasing q via the expectation quality in (5).

A. Randomized Trials

We performed a set of simulated exploration trials, where each trial consisted of the six sampling strategies sequentially selecting observation locations and searching for the best path over a 2D cost field on a 100×100 square grid (as shown in Fig. 5). Trials do not have any observations at the start ($Z_0 = \emptyset$). Since FMEx scales roughly linearly with p and q, we set the number of sample locations at $p = 50$ for both FMEx3σ and FMExMC6, and $p = 150$ for FMEx1σ to provide equal computation time for FMEx variants. Sample locations were selected randomly from available grid points. The comparison methods (Max Var and LCB) are low computational cost, because they only require finding a maximum ($\mathcal{O}(n)$). To account for this, we allowed both methods to sample densely from the grid ($p = 10000$). Although this still results in a lower computation

time than FMEx, we do not believe that further increasing their sample density would improve performance.

In each trial, a 'true' cost field f was generated by summing the contributions of a set of randomized circular Gaussian peaks. Each component was characterized by a peak value h_j, radius r_j and center location $\mathbf{c} = (x_j, y_j)$, as shown in (9). Peak values and radii were randomly sampled from uniform distributions in the intervals $[-3.0, 8.0]$ and $[5.0, 12.0]$ respectively, and a base value $c_0 = 5.0$.

$$f(\mathbf{x}) = c_0 + \sum_j h_j \exp(-\|\mathbf{x} - \mathbf{c}\|/r_j) \tag{9}$$

We ran two sets of experiments, with and without (known) fixed obstacles, each for 100 randomized trials. In obstacle trials, we randomly placed 10 rectangular regions with side lengths drawn from $U(2, 10)$. The start position $\mathbf{x}_s = (3, 3)$ and $\mathbf{x}_g = (97, 97)$. Simulated sensor observations were sampled from $\mathcal{N}(f(\mathbf{x}), 0.01)$. The GP used a fixed length scale of 10.0 and variance 15.0. Each iteration, FMEx methods took approximately 4 s each to select an observation location, and Max Var and LCB took ~ 0.5 s.

Fig. 5 shows a randomized trial at $t = 30$. Max variance samples uniformly to minimize the uncertainty evenly across the entire field. LCB sampling tends to sample more densely in low-cost regions and sparsely elsewhere. Although the FMEx3σ method did not necessarily model the complete cost field well (see the lower right region), it places samples to maximize the likelihood of finding the best path, such that promising areas are sampled more densely than areas that are unlikely to contain a lower-cost path.

Fig. 6 shows the additional path cost (compared to the optimal path) of the path predicted by each method against the number of observations. The true cost C_t of each predicted path is calculated by taking the path generated by each method, $\mathcal{P}_m$, and calculating the cost of that path on the true cost field f using numerical integration along the path. The true best path cost is C_{best}. The additional path cost is the relative additional cost of the predicted path versus the true best path $(C_t(\mathcal{P}_m) - C_{\text{best}})/C_{\text{best}} - 1$.

Lines for each method represent median scores across 100 randomized trials, and vertical bars show the extent of the first and third quartiles. We show quartiles rather than standard deviation because the data is not balanced above and below the median. In the first 10–15 samples none of the methods have an adequate map of the environment and hence path costs are relatively high. As more samples are collected, both FMEx3σ and 1σ find consistently better paths than LCB or max variance. However, the MC6 sampler performs relatively poorly. We think that the balanced sampling using the fixed σ increments results in more consistent estimation of the path cost expectation, whereas the random Monte Carlo sampling has too few samples ($q = 6$) to produce consistently good estimates. Increasing q for FMExMC could potentially improve performance at the cost of computation.

Fig. 7 shows how well each method estimates the true path cost. The vertical axis shows the normalized RMS error over all randomized trials between the estimated cost $C_e(\mathcal{P}_m)$ of the predicted best path from each model, and the true cost of the path from the actual cost field, $C_t(\mathcal{P}_m)$. With fewer samples

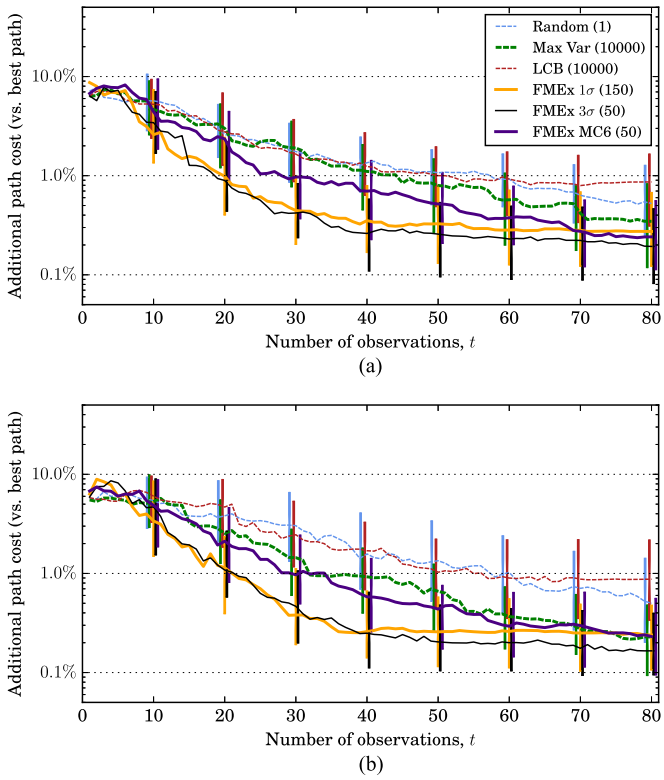

Fig. 6. Additional path cost relative to optimal over 100 randomized trials. Lines show median values, vertical bars cover first and third data quartiles. Parentheses in the legend indicate the number of locations each method chooses from at each iteration. (a) No obstacles. (b) Obstacles.

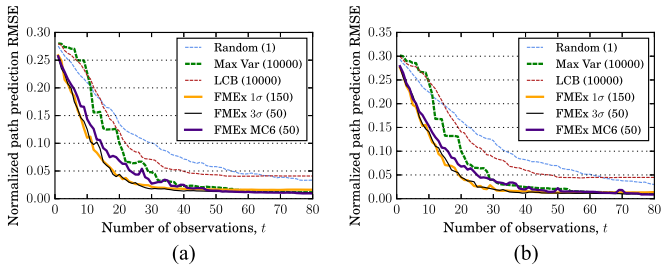

Fig. 7. Path cost prediction error (RMS) over 100 randomized trials.

FMEx generates more accurate path cost estimates. Random and maximum variance sampling catch up in path cost and cost estimation performance as the number of samples becomes sufficient for good coverage of the search space, in which case the sampling method becomes less important as long as it provides coverage.

B. Sample Planning From Ocean Bathymetry

To demonstrate the ability of FMEx to model more complex domains, we provide results for an ocean sampling problem. We considered the problem of selecting sample locations to find a path for a submarine cable from Iceland to the United Kingdom. We used bathymetry from satellite surveys[1], subsampled to a grid resolution of 240×132, and re-scaled to a range $(0.05, 1.0)$

[1]http://coastwatch.pfeg.noaa.gov/erddap/griddap

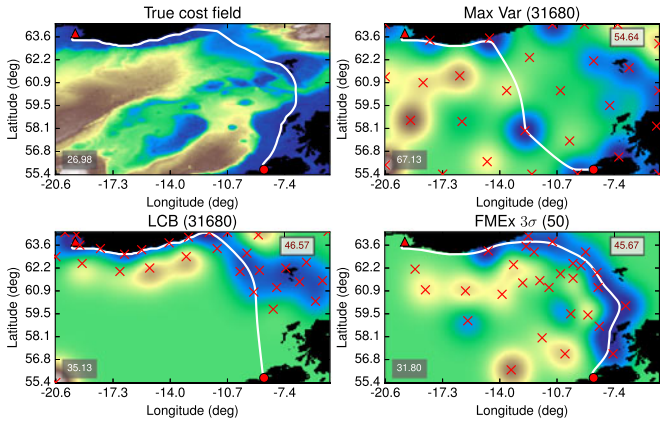

Fig. 8. Cost field and path estimates after $t = 30$ samples on bathymetry data (blue is shallower). For each sampling method the true path cost is shown in the bottom left (white text) and estimated path cost in the top right (red text) of each subfigure.

corresponding to depths $(0, 3800)$ m to generate the 'true' cost field f. The cost of sampling is assumed to be high as it involves deploying a submersible to survey the sea floor at each location. Fig. 8 illustrates the cost map estimates, best paths, and path costs (true and estimated) after $t = 30$. In this case, we can see that both LCB and FMEx3σ found relatively good paths after 30 samples, though FMEx provided a better estimate of the path cost. Max Var sampling performed poorly as it had not yet sampled densely enough in some important regions.

C. Discussion

FMEx requires user-defined estimates for the grid resolution, values of p and q, and for the GP mean and hyperparameters. The fixed 3σ sampling ($q = 6$) showed the best performance in our tests. Since FMEx is effectively an anytime algorithm with respect to sample locations, it could be implemented to test locations until a computational time budget is filled during each iteration, and return the current best estimate. FMEx is also sensitive to the GP hyperparameters. If the model variance is too low, and the true solution lies outside what the GP prior expects, then the sampling can become stuck in local minima. However, we did not find this to be a significant problem in our tests with reasonable hyperparameter estimates, and FMEx found a low-cost path in a majority of cases.

VI. Conclusion

This paper presented the FMEx algorithm for sequentially selecting observations to sample a cost map to find the least-cost path. FMEx is based on a probabilistic model of the cost field generated using GP regression from previous observations. Using the probabilistic output of the GP, the FMEx algorithm generates an approximation of the expected best path cost acquired by sampling a target location. The proposed biFM path cost update provides efficient calculation of path cost changes caused by localized map updates, requiring significantly fewer node expansions than standard FM or E*. Using the biFM path cost estimate to approximate the path cost expectation over possible samples, FMEx successively selects observations that are likely to reduce path cost. Simulation results show that FMEx generates sample locations that improve path quality by finding lower-cost paths and more accurate estimates of the true path cost with fewer samples compared to random, maximum variance or confidence bound sampling.

VII. Future Work

A feature that we would like to pursue for FMEx is automated termination using path quality estimates from the GP variance based on a desired path accuracy. We would also like to investigate alternative observation selection models and anisotropic costs to take into account the dynamics and traversal costs of the exploring vehicle.

References

[1] C. H. Elmendorf and B. C. Heezen, "Oceanographic information for engineering submarine cable systems," *Bell Syst. Tech. J.*, vol. 36, no. 5, pp. 1035–1094, Sep. 1957.
[2] D. M. Doolin and N. Sitar, "Wireless sensors for wildfire monitoring," *Proc. SPIE*, vol. 5765, pp. 477–484, 2005.
[3] P. E. I. Pounds and S. P. N. Singh, "Integrated electro-aeromechanical structures for low-cost, self-deploying environment sensors and disposable UAVs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 4459–4466.
[4] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, pp. 1591–1595, 1996.
[5] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Jun. 2008.
[6] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *J. Artif. Intell. Res.*, vol. 42, no. 1, pp. 427–486, 2011.
[7] A. Ghaffarkhah and Y. Mostofi, "Communication-aware motion planning in mobile networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2478–2485, Oct. 2011.
[8] J. J. Chung, N. R. J. Lawrance, and S. Sukkarieh, "Learning to soar: Resource-constrained exploration in reinforcement learning," *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 158–172, 2015.
[9] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 2242–2249.
[10] J. A. Sethian and A. Vladimirsky, "Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms," *SIAM J. Numerical Anal.*, vol. 41, no. 1, pp. 325–363, 2003.
[11] S. Garrido, L. Moreno, M. Abderrahim, and F. Martin, "Path planning for mobile robot navigation using Voronoi diagram and Fast Marching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2376–2381.
[12] C. Pêtrès, Y. Pailhas, P. Patrón, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 331–341, Apr. 2007.
[13] J. Elston and E. W. Frew, "Unmanned aircraft guidance for penetration of pretornadic storms," *J. Guidance, Control, Dyn.*, vol. 33, no. 1, pp. 99–107, 2010.
[14] J. V. Gómez, A. Vale, S. Garrido, and L. Moreno, " Performance analysis of fast marching-based motion planning for autonomous mobile robots in ITER scenarios," *Robot. Auton. Syst.*, vol. 63, pp. 36–49, 2015.
[15] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The Field D* algorithm," *J. Field Robot.*, vol. 23, no. 2, pp. 79–101, 2006.
[16] R. Philippsen and R. Siegwart, "An interpolated dynamic navigation function," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 3782–3789.
[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Series Adaptive Computation and Machine Learning.)* Cambridge, MA, USA: MIT Press, 2006.
[18] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. 13th Int. Conf. Very Large Data Bases-Vol. 30*, 2004, pp. 588–599.