Accuracy-Aware Aquatic Diffusion Process Profiling Using Robotic Sensor Networks

Yu Wang*, Rui Tan*, Guoliang Xing*, Jianxun Wang†, and Xiaobo Tan†
*Department of Computer Science and Engineering, Michigan State University, USA
†Department of Electrical and Computer Engineering, Michigan State University, USA
{wangyu3, tanrui, glxing, wangji19, xbtan}@msu.edu

ABSTRACT

Water resources and aquatic ecosystems are facing increasing threats from climate change, improper waste disposal, and oil spill incidents. It is of great interest to deploy mobile sensors to detect and monitor certain diffusion processes (e.g., chemical pollutants) that are harmful to aquatic environments. In this paper, we propose an accuracy-aware diffusion process profiling approach using smart aquatic mobile sensors such as robotic fish. In our approach, the robotic sensors collaboratively profile the characteristics of a diffusion process including source location, discharged substance amount, and its evolution over time. In particular, the robotic sensors reposition themselves to progressively improve the profiling accuracy. We formulate a novel movement scheduling problem that aims to maximize the profiling accuracy subject to limited sensor mobility and energy budget. We develop an efficient *greedy* algorithm and a more complex near-optimal *radial* algorithm to solve the problem. We conduct extensive simulations based on real data traces of robotic fish movement and wireless communication. The results show that our approach can accurately profile dynamic diffusion processes under tight energy budgets. Moreover, a preliminary evaluation based on the implementation on TelosB motes validates the feasibility of deploying our movement scheduling algorithms on mote-class robotic sensor platforms.

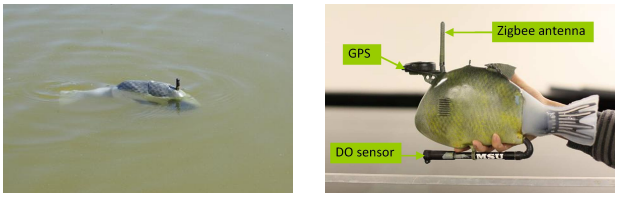
Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Signal processing systems; C.4 [**Performance of Systems**]: Measurement techniques, modeling techniques; G.1.6 [**Numerical Analysis**]: Optimization—*Constrained optimization, gradient methods*

Keywords

Robotic sensor networks, diffusion process, movement scheduling

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IPSN'12, April 16–20, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1227-1/12/04 ...$10.00.



(a) (b)

Figure 1: Prototypes of autonomous robotic fish developed by the Smart Microsystems Laboratory at Michigan State University [26].

1. INTRODUCTION

Water resources and aquatic ecosystems have been facing various physical, chemical, and biological threats from climate change, industrial pollution, and improper waste disposal. For instance, the last four decades witnessed more than a dozen major oil spills with each releasing more than 30 million gallons of oil [1]. Other harmful diffusion processes like chemical or radiation leaks could also have disastrous impact on public health and ecosystem sustainability. When such a crisis arises, an immediate requirement is to profile the characteristics of the diffusion process, including the location of source, the amount of discharged substance, and how rapidly it spreads in space and evolves over time.

Manual sampling, via boat/ship or with handhold devices, is still a common practice in the monitoring of aquatic diffusion processes. This approach is labor-intensive and difficult to adapt to the dynamic evolution of diffusion. An alternative is *in situ* sensing with fixed or buoyed/moored sensors [20]. However, since buoyed sensors cannot move around, it could take a prohibitively large number of them to capture spatially inhomogeneous information. The past couple of decades have seen significant progress in developing robotic technologies for aquatic sensing. Autonomous underwater vehicles (AUVs) [10] and sea gliders [9] are notable examples of such technologies. However, because of their high cost (over 50,000 U.S. dollars per unit [21]), weight (over 100 pounds), and size (1-2 meters long), it is difficult to deploy many AUVs or sea gliders for temporally and spatially resolved measurement of diffusion processes.

Recent advances in computing, communication, sensing, and actuation technologies have made it possible to create untethered *robotic fish* with onboard power, control, navigation, wireless communication, and sensing modules, which turn these robots into mobile sensing platforms in aquat-

ic environments. Fig. 1(a) shows a prototype of robotic fish swimming in an inland lake. Fig. 1(b) shows the close-up of a robotic fish prototype, equipped with GPS, Zigbee antenna, and a dissolved oxygen (DO) sensor. Due to the low manufacturing cost, these platforms can be massively deployed to form a mobile sensor network that monitors harmful diffusion processes, providing significantly higher spatial and temporal sensing resolution than existing monitoring methods. Moreover, a school of robotic fish can coordinate their sensing and movements through wireless communication enabled by the onboard Zigbee radio, to adapt to the dynamics of evolving diffusion processes.

Despite the aforementioned advantages, low-cost mobile sensing platforms like robotic fish introduce several challenges for aquatic sensing. First, due to the constraints on size and energy, they are typically equipped with low-end sensors whose measurements are subject to significant biases and noises. They must efficiently collaborate in data processing to achieve satisfactory accuracy in diffusion profiling. Second, practical aquatic mobile platforms are only capable of relatively low-speed movements. Hence the movements of sensors must be efficiently scheduled to achieve real-time profiling of the diffusion processes that may evolve rapidly over time. Third, given the high power consumption of locomotion, the distance that mobile sensors move in a profiling process should be minimized to extend the network lifetime.

We make the following major contributions in this paper:

- We propose a novel accuracy-aware approach for aquatic diffusion profiling based on robotic sensor networks. Our approach leverages the mobility of robotic sensors to iteratively profile the spatiotemporally evolving diffusion process.
- We derive the analytical profiling accuracy of our approach based on the Cramér-Rao bound (CRB). Then we formulate a movement scheduling problem for aquatic diffusion profiling, in which the profiling accuracy is maximized under the constraints on sensor mobility and energy budgets. We develop gradient-ascent-based *greedy* and dynamic-programming-based *radial* movement scheduling algorithms to solve the problem.
- We implement the profiling and movement scheduling algorithms on TelosB motes and evaluate the system overhead. Moreover, we conduct extensive simulations based on real data traces of robotic fish movement and wireless communication for evaluation. The results show that our approach can accurately profile the dynamic diffusion process and adapt to its evolution.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the preliminaries and Section 4 provides an overview of our approach. Section 5 derives the analytical profiling accuracy metric and Section 6 formulates the movement scheduling problem. Section 7 presents the two movement scheduling algorithms. Section 8 discusses several extensions. Section 9 presents evaluation results and Section 10 concludes this paper.

2. RELATED WORK

Most previous work on diffusion process monitoring is based on stationary sensor networks. Several different estimation techniques are adopted by these studies, which include *state-space filtering*, *statistical signal processing*, and *geometric trilateration*. The state space approach [19, 31] uses discrete state-space equations to approximate the partial differential equations that govern the diffusion process, and then applies filtering algorithms such as Kalman filters [19, 31] to profile the diffusion process based on noisy measurements. In the statistical signal processing approach, several estimation techniques such as MLE [17, 29] and Bayesian parameter estimation [33] are applied to deal with noisy measurements. For instance, in [29], an MLE-based diffusion characterization algorithm is designed based on binary sensor measurements to reduce the communication cost. In [33], the parametric probability distribution of the diffusion profile parameters is passed among sensors and updated with sensor measurements by a Bayesian estimation algorithm. The passing route is determined according to various estimation performance metrics including CRB. In geometric trilateration approach [14, 4], the measurement of a sensor is mapped to the distance from the sensor to the diffusion source. The source location can then be estimated by trilateration among multiple sensors. Such an approach incurs low computational complexities, but suffers lower estimation accuracy compared with more advanced approaches such as MLE [4].

Recently, sensor mobility has been exploited to enhance the adaptability and sensing capability of sensor networks. For instance, heuristic movement scheduling algorithms are proposed in [24] to estimate the contours of a physical field. In [22], more complex path planning schemes are proposed for mobile sensors to reconstruct a spatial map of environmental phenomena that do not follow a particular physical model. Our previous works exploit reactive sensor mobility to improve the detection performance of a sensor network [25, 32]. Several studies are focused on using robots to improve the accuracy in profiling diffusion processes. As an extension to [17], the gradient of CRB is used to schedule the movement of a single sensor in [29]. Similarly, a robot motion control algorithm is proposed in [23] to maximize the determinant of the Fisher information matrix. However, as these diffusion profiling approaches [17, 23, 29] adopt complicated numerical optimization, they are only applicable to a small number (e.g., 3 in [23]) of powerful robots. In contrast, we focus on developing movement scheduling algorithms for moderate- or large-scale mobile networks that are composed of inexpensive robotic sensors.

3. PRELIMINARIES

In this section, we describe the preliminaries including the diffusion process and sensor models.

3.1 Diffusion Process Model

A diffusion process in a static aquatic environment, by which molecules spread from areas of higher concentration to areas of lower concentration, follows Fick's law [13]. In addition to the diffusion, the spread of the discharged substance is also affected by the advection of solvent, e.g., the movement of water caused by the wind. By denoting t as the time elapsed since the discharge of substance and c as the substance concentration, the diffusion-advection model can be described as

$$\frac{\partial c}{\partial t} = D_x \cdot \frac{\partial^2 c}{\partial x^2} + D_y \cdot \frac{\partial^2 c}{\partial y^2} + D_z \cdot \frac{\partial^2 c}{\partial z^2} - u_x \cdot \frac{\partial c}{\partial x} - u_y \cdot \frac{\partial c}{\partial y}, \quad (1)$$

where D is the diffusion coefficient, u is the advection speed,

and the subscripts of D and u denote the directions (i.e., x-, y-, and z-axis). The diffusion coefficients characterize the speed of diffusion and depend on the species of solvent and discharge substance as well as other environment factors such as temperature. The advection speeds characterize the horizontal solvent movement caused by external forces such as wind and flow. The above Fickian diffusion-advection model has been widely adopted to study the spreading of gaseous substances [29] and buoyant fluid pollutants such as oil slick on the sea [16]. For many buoyant fluid pollutants, the two horizontal diffusion coefficients, i.e., D_x and D_y, are identical, while the vertical diffusion coefficient, i.e., D_z, is insignificant. For instance, in a field experiment [8], where diesel oil was discharged into the sea, the estimated D_x is 2,000 cm^2/s while D_z is only 10 cm^2/s. Therefore, the vertical diffusion coefficient can be safely ignored and the diffusion can be well characterized by a 2-dimensional process. In this paper, our study is focused on buoyant fluid pollutants with the diffusion coefficients $D_x = D_y = D$.

Suppose a total of $A\,\mathrm{cm}^3$ of substance is discharged at location (x_s, y_s) and $t = 0$. At time $t > 0$, the original diffusion source is drifted to (x_0, y_0) due to advection, where $x_0 = x_s + u_x t$ and $y_0 = y_s + u_y t$. Hereafter, by *source location* we refer to the source location that has *drifted* from the original position due to advection, unless otherwise specified. Denote $d(x, y)$ (abbreviated to d) as the distance from any location (x, y) to the source location, i.e., $d = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. In the presence of advection, the diffusion is isotropic with respect to the drifted source location [6]. Therefore, the concentration at (x, y) can be denoted as $c(d, t)$. The initial condition for Eq. (1) is an impulse source, which can be represented by the Dirac delta function, i.e., $c(d, 0) = A \cdot \delta(d)$. Under this initial condition, the closed-form solution to Eq. (1) is given by [29]:

$$c(d,t) = \alpha \cdot \exp\left(-\beta \cdot d^2\right), \quad d \geq 0, t > 0, \tag{2}$$

where $\alpha = \frac{A}{4\pi D t}$ and $\beta = \frac{1}{4Dt}$. From Eq. (2), for a given time instant t, the concentration distribution is described by the Gaussian function that centers at the source location. As time elapses, the concentration distribution becomes flatter. In this paper, the *diffusion profile* is defined as $\Theta = \{x_0, y_0, \alpha, \beta\}$.

3.2 Sensor Model

Our approach leverages mobile nodes (e.g., robotic fish [26]) to collaboratively profile an aquatic diffusion process. The nodes form a cluster and a cluster head is selected to process the measurements from cluster members. The selection of cluster head will be discussed in Section 7.2. Moreover, we will extend our approach to address multiple clusters in Section 8.2. Many aquatic mobile platforms are battery-powered and hence have limited mobility and energy budget. For instance, the movement speed of the robotic fish designed in [26] was about 1.8 to 6 m/min. We assume that the mobile nodes are equipped with pollutant concentration sensors (e.g., the Cyclops-7 [28] series) that can measure the concentrations of crude oil, harmful algae, etc. Lastly, we assume that the sensors are equipped with low-power wireless interfaces (e.g., 802.15.14 ZigBee radios) and hence can communicate with each other when on water surface.

The measurements of most sensors are subject to biases and additive random noises from the sensor circuitry and the environment. Specifically, the reading of sensor i, denoted by z_i, is given by $z_i = c(d_i, t) + b_i + n_i$, where d_i is the distance from sensor i to the diffusion source, b_i and n_i are the bias and random noise for sensor i, respectively. In the presence of constant-speed advection, the source and the sensors will drift with the same speed and therefore they are in the same inertial system. As a result, the concentration at the position of sensor i is given by $c(d_i, t)$. We assume that the noise experienced by sensor i follows the zero-mean normal distribution with variance ς^2, i.e., $n_i \sim \mathcal{N}(0, \varsigma^2)$. We assume that the noises, i.e., $\{n_i | \forall i\}$, are independent across sensors. The bias and noise variance for calm water environment are often given in the sensor specification provided by the manufacturer. They may also be measured in offline lab experiments. For instance, by placing a sensor in the pollutant-free fluid media, the bias and noise variance can be estimated by the sample mean and variance over a number of readings. When the water environment is wavy, the noise variance will increase. Therefore, to address wavy environment, the noise variance should be measured in offline lab experiments with various wavy levels. The above measurement model has been widely adopted for various chemical sensors [17, 29, 33].

In this paper, we adopt a temporal sampling scheme to mitigate the impact of noise. Specifically, when sensor i measures the concentration, it continuously takes K samples in a short time, and computes the average as its measurement. Therefore, the measurement z_i follows the normal distribution, i.e., $z_i \sim \mathcal{N}(c(d_i, t) + b_i, \sigma^2)$, where $\sigma^2 = \varsigma^2 / K$.

Table 1 summarizes the notation used in this paper.

Table 1: Summary of Notation

Symbol	Definition
D	diffusion coefficient
A	total amount of discharged substance in cm^3
t	time from the discharge of substance
α, β	$\alpha = A(4\pi Dt)^{-1}$, $\beta = (4Dt)^{-1}$
(x_s, y_s)	coordinates of the original diffusion source
(x_0, y_0)	coordinates of the drifted diffusion source
$(x_i{+}x_0, y_i{+}y_0)$	coordinates of sensor i
d_i	distance from the drifted diffusion source
$c(d_i, t)$	concentration at sensor i and time t
Θ	diffusion process profile $\Theta = \{x_0, y_0, \alpha, \beta\}$
b_i, σ^2	sensor bias and noise variance
n_i	Gaussian noise, $n_i \sim \mathcal{N}(0, \sigma^2)$
z_i	sensor measurement, $z_i \sim \mathcal{N}(c(d_i, t) + b, \sigma^2)$
K	number of samples for computing a measurement
N	total number of sensors
$\mathbf{z}$	normalized observation, $\mathbf{z} = [\frac{z_1 - b_1}{\sigma}, \ldots, \frac{z_N - b_N}{\sigma}]^{\mathrm{T}}$
ω	diffusion process profiling accuracy metric
v	sensor movement speed

* The symbols with subscript i refer to the notation of sensor i.

4. OVERVIEW OF APPROACH

In this section, we provide an overview of our approach. Our objective is to profile (i.e., estimate Θ) of an aquatic diffusion-advection process using a robotic sensor network. Our approach is designed to meet two key objectives. First, the noisy measurements of sensors are jointly processed to improve the accuracy in profiling the diffusion. Second, sensors can actively move based on current measurements to maximize the profiling accuracy subject to the energy consumption budget. With the estimated profile $\widetilde{\Theta}$, we can learn several important characteristics of the diffusion pro-

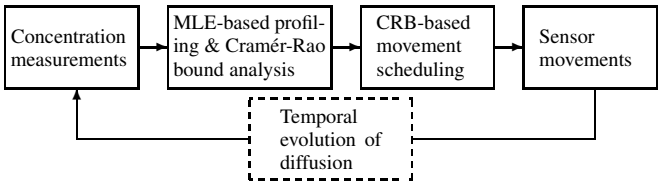


Figure 2: The iterative diffusion profiling process.

cess of interest.[1] First, we can compute the current concentration contour maps with Eq. (2). Second, we can estimate the elapsed time since the start of the diffusion and the total amount of discharged substance, with $\widetilde{t} = (4D\widetilde{\beta})^{-1}$ and $\widetilde{A} = \pi\widetilde{\alpha}\widetilde{\beta}^{-1}$. Third, we can estimate the original source location by $\widetilde{x}_s = \widetilde{x}_0 - u_x\widetilde{t}$ and $\widetilde{y}_s = \widetilde{y}_0 - u_y\widetilde{t}$. Moreover, we can predict the evolution of the diffusion in the future, which is often important for emergency management in the cases of harmful substance discharge.

We assume that the robotic sensors are initially distributed at randomly chosen positions in the deployment region that covers the diffusion source. For instance, the sensors can be dropped off from an unmanned aerial vehicle or placed by an aquatic vessel randomly. Note that random and sometimes uniform deployment of sensors around the source location is a good strategy when the characteristics of diffusion process have yet to be determined. This also avoids the massive locomotion energy required to spread sensors for a satisfactory profiling accuracy when the diffusion process evolves. We assume that all sensors know their positions (e.g., through GPS or an in-network localization service) and are time-synchronized. In Section 9.3.6, we will evaluate the impact of initial sensor deployment on the profiling accuracy and locomotion energy consumption of our approach.

After the initial deployment, sensors begin a diffusion profiling process consisting of multiple *profiling iteration*s. The iterative profiling process is illustrated in Fig. 2. In a profiling iteration, sensors first simultaneously take concentration measurements and send them to the cluster head, using a possibly multi-hop wireless communication protocol. The cluster head then adopts the maximum likelihood estimation (MLE) to estimate Θ from the noisy measurements of all sensors. With the estimated diffusion profile, the cluster head schedules sensor movements such that the expected profiling accuracy in the next profiling iteration is maximized, subject to limited sensor mobility and energy budget. Finally, the movement schedule including moving orientations and distances is sent to sensors for directing their movements.

Our accuracy-aware diffusion profiling approach features the following novelties. First, it starts with little prior knowledge about the diffusion and progressively learns the profile of the diffusion with improved accuracy along the profiling iterations. As sensors resample the concentration in each iteration, such an iterative profiling strategy allows the network to adapt to the dynamics of the diffusion process while reducing energy consumption of robotic sensors. Moreover, although the profiling accuracy in each iteration is affected by errors in sensor localization and movement control, our approach only schedules short-distance movements for sensors in each iteration and updates their positions in the next iteration, which avoids the accumulation of errors in sensor localization and movement control. Second, we analyze the CRB of the MLE-based diffusion profiling algorithm and propose a novel CRB-based profiling accuracy metric, which is used to direct the movement scheduling of robotic sensors. Third, we propose two novel movement scheduling algorithms, which include a gradient-ascent-based *greedy* algorithm and a dynamic-programming-based *radial* algorithm. The *greedy* algorithm only incurs linear complexity, while the *radial* algorithm can find the near-optimal movement schedule with a higher but still polynomial complexity.

[1]For the clarity of presentation, we denote $\widetilde{x}$ as the estimate of x.

5. PROFILING ALGORITHM AND ACCURACY ANALYSIS

In this section, we first present our MLE-based diffusion profiling algorithm, which estimates the diffusion profile Θ in each profiling iteration. We then analyze the theoretical profiling accuracy based on Cramér-Rao bound (CRB). The closed-form relationship between the profiling accuracy and the sensors' positions will guide the design of our accuracy-aware sensor movement scheduling algorithms.

5.1 MLE-based Diffusion Profiling Algorithm

MLE and Bayesian parameter estimation are two typical parameter estimation approaches [7]. The Bayesian estimation relies on prior probability distribution of the parameters, which is often unknown and difficult to model in practice. In this paper, we adopt MLE to estimate the profile of the diffusion process. Specifically, we assume that N aquatic sensors are deployed in the region of interest. In each profiling iteration, we first remove sensor biases and normalize the measurements to construct the observation vector $\mathbf{z}$, which is given by $\mathbf{z} = [\frac{z_1-b_1}{\sigma}, \ldots, \frac{z_N-b_N}{\sigma}]^{\mathrm{T}}$. By denoting $\mathbf{H} = [\sigma^{-1}\mathrm{e}^{-\beta d_1^2}, \ldots, \sigma^{-1}\mathrm{e}^{-\beta d_N^2}]^{\mathrm{T}}$, $\mathbf{z}$ follows the N-dimensional normal distribution, i.e., $\mathbf{z} \sim \mathcal{N}(\alpha\mathbf{H}, \mathbf{I})$, where $\mathbf{I}$ is the $N \times N$ identity matrix. The log-likelihood of an observation $\mathbf{z}$ given Θ is given by [7]:

$$\mathcal{L}(\mathbf{z}|\Theta) = -(\mathbf{z} - \alpha\mathbf{H})^{\mathrm{T}}(\mathbf{z} - \alpha\mathbf{H}). \quad (3)$$

MLE aims to maximize the log-likelihood given by Eq. (3). Formally, $\widetilde{\Theta}(\mathbf{z}) = \arg\max_{\Theta} \mathcal{L}(\mathbf{z}|\Theta)$. This unconstrained optimization problem can be solved by various numerical methods, e.g., Nelder-Mead's algorithm [18].

5.2 Cramér-Rao Bound for Diffusion Profiling

CRB provides a theoretical lower bound on the variance of parameter estimators [7], and has been widely adopted to guide the design of estimation algorithms [17, 33]. This section derives the CRB of profile Θ estimation. CRB is given by the inverse of the Fisher information matrix (FIM) [7]. For the diffusion profiling, the FIM is defined by $\mathbf{J} = -\mathbb{E}\left[\frac{\partial}{\partial\Theta}\left(\frac{\partial}{\partial\Theta}\mathcal{L}(\mathbf{z}|\Theta)\right)\right] = \alpha^2\frac{\partial\mathbf{H}^{\mathrm{T}}}{\partial\Theta}\frac{\partial\mathbf{H}}{\partial\Theta}$, where the expectation $\mathbb{E}[\cdot]$ is taken over all possible $\mathbf{z}$. The k^{th} diagonal element of the inverse of $\mathbf{J}$ (denoted by $\mathbf{J}^{-1}_{k,k}$) provides the lower bound on the variance of the k^{th} element of $\widetilde{\Theta}$ (denoted by $\widetilde{\Theta}_k$) [7]. Formally, $\mathrm{Var}(\widetilde{\Theta}_k) \geq \mathbf{J}^{-1}_{k,k}$. The number $\mathbf{J}^{-1}_{k,k}$ is the CRB corresponding to Θ_k, which is denoted as $\mathrm{CRB}(\Theta_k)$ in this paper. Although the CRB can be easily computed

via numerical methods, in order to guide the movements of sensors, we will derive the closed-form CRB.

Even though $\mathbf{J}$ is just a 4×4 matrix, deriving $\mathbf{J}^{-1}$ is challenging, because the N-dimensional joint distribution function in Eq. (3) leads to high inter-node dependence. To simplify the discussion, we set up a Cartesian coordinate system with the origin at the source location and let (x_i, y_i) denote the coordinates of sensor i. Note that the coordinates of the diffusion source and sensor i in the global coordinate system are (x_0, y_0) and $(x_0 + x_i, y_0 + y_i)$, respectively. We apply matrix calculus to derive the closed-form $\mathbf{J}$ and then derive $\mathbf{J}^{-1}$ by block matrix manipulations. Due to space limitation, the details of the derivations are omitted here and can be found in [30]. To facilitate the representation of CRB, we first define several notations, i.e., $\widehat{x}_i$, $\widehat{y}_i$, $\mathbf{L}_{X_1}$, $\mathbf{L}_{X_2}$, $\mathbf{L}_{Y_1}$, and $\mathbf{L}_{Y_2}$. First, $\widehat{x}_i$ is given by

$$\widehat{x}_i = \frac{\sum_{j=1}^{N} x_j (d_j^2 - d_i^2) \mathrm{e}^{-2\beta d_j^2}}{\sqrt{\sum_{m=1}^{N} \sum_{n=1}^{N} (d_m^2 - d_n^2)^2 \mathrm{e}^{-2\beta (d_m^2 + d_n^2)}}}. \quad (4)$$

By replacing x_j in Eq. (4) with y_j, we can define $\widehat{y}_i$ in a similar manner. Moreover, $\mathbf{L}_{X_1}$, $\mathbf{L}_{Y_1}$ are $1 \times N$ vectors, and $\mathbf{L}_{X_2}$, $\mathbf{L}_{Y_2}$ are $N \times 1$ vectors. The i^{th} elements of them are

$$\mathbf{L}_{X_1}(i) = \sigma^{-1} \mathrm{e}^{-\beta d_i^2} (x_i + \widehat{x}_i), \quad \mathbf{L}_{Y_1}(i) = \sigma^{-1} \mathrm{e}^{-\beta d_i^2} (y_i + \widehat{y}_i),$$
$$\mathbf{L}_{X_2}(i) = \sigma^{-1} \mathrm{e}^{-\beta d_i^2} (x_i - \widehat{x}_i), \quad \mathbf{L}_{Y_2}(i) = \sigma^{-1} \mathrm{e}^{-\beta d_i^2} (y_i - \widehat{y}_i).$$

Based on the above notation, the CRBs for the estimates of x_0 and y_0 are given by

$$\mathrm{CRB}(x_0) = \mathbf{J}_{1,1}^{-1} = \frac{(4\alpha^2\beta^2)^{-1}}{\mathbf{L}_{X_1}\mathbf{L}_{X_2} - \frac{\left(\mathbf{L}_{X_1}\mathbf{L}_{Y_2} + \mathbf{L}_{X_2}\mathbf{L}_{Y_1}\right)^2}{4\mathbf{L}_{Y_1}\mathbf{L}_{Y_2}}}, \quad (5)$$

$$\mathrm{CRB}(y_0) = \mathbf{J}_{2,2}^{-1} = \frac{(4\alpha^2\beta^2)^{-1}}{\mathbf{L}_{Y_1}\mathbf{L}_{Y_2} - \frac{\left(\mathbf{L}_{X_1}\mathbf{L}_{Y_2} + \mathbf{L}_{X_2}\mathbf{L}_{Y_1}\right)^2}{4\mathbf{L}_{X_1}\mathbf{L}_{X_2}}}. \quad (6)$$

5.3 Profiling Accuracy Metric

In this section, we propose a novel diffusion profiling accuracy metric based on the CRBs derived in Section 5.2, which will be used to guide the movements of sensors in Section 6. Several previous works [23] adopt the determinant of the FIM as the accuracy metric, which jointly considers all the parameters. Such a metric requires the parameters to be properly normalized to avoid biases. However, normalizing the parameters with different physical meanings is highly problem-dependent. Moreover, as the closed-form determinant of the FIM is extremely complicated, the resulted sensor movement scheduling has to rely on the numerical methods with high computational complexities [23], which is not suitable for robotic sensors with limited resources. In this paper, we propose a new profiling accuracy metric, denoted by ω, which is defined according to the sum of reciprocals of $\mathrm{CRB}(x_0)$ and $\mathrm{CRB}(y_0)$. Formally,

$$\omega = \frac{\frac{1}{\mathrm{CRB}(x_0)} + \frac{1}{\mathrm{CRB}(y_0)}}{4\alpha^2\beta^2} = (1-\epsilon)\left(\mathbf{L}_{X_1}\mathbf{L}_{X_2} + \mathbf{L}_{Y_1}\mathbf{L}_{Y_2}\right), \quad (7)$$

where $\epsilon = \frac{\left(\mathbf{L}_{X_1}\mathbf{L}_{Y_2} + \mathbf{L}_{X_2}\mathbf{L}_{Y_1}\right)^2}{4\mathbf{L}_{X_1}\mathbf{L}_{X_2}\mathbf{L}_{Y_1}\mathbf{L}_{Y_2}}$. By adopting reciprocals, the accuracy analysis can be greatly simplified. Note that as α and β are unknown but fixed in a particular profiling iteration, $4\alpha^2\beta^2$ in the denominator of Eq. (7) is a scaling factor. Therefore, optimizing $\frac{1}{\mathrm{CRB}(x_0)} + \frac{1}{\mathrm{CRB}(y_0)}$ is equivalent to optimizing ω. As discussed in Section 5.1, we adopt the MLE to estimate Θ. The variance of the MLE result converges to CRB and hence a larger ω indicates more accurate estimation of x_0 and y_0. With the metric ω, the movements of sensors will be directed according to the accuracy of localizing the diffusion source. In the rest of this paper, the term *profiling accuracy* refers to the metric ω defined in Eq. (7). Note that our approach can also be applied to focus on the profiling accuracy of the elapsed time t and discharged substance amount A, by applying the same matrix manipulations to obtain $\mathrm{CRB}(\alpha)$ and $\mathrm{CRB}(\beta)$.

According to the derivations in Section 5.2, $\mathbf{L}_{X_1}$, $\mathbf{L}_{Y_1}$, $\mathbf{L}_{X_2}$ and $\mathbf{L}_{Y_2}$ depend on x_0, y_0, x_i and y_i. Therefore, ω is a function of the positions of the sensors and the diffusion source. As the location of the diffusion source, i.e., (x_0, y_0), is unknown to the network, it is impossible to compute the true profiling accuracy. In our approach, we compute ω based on the estimated location of the diffusion source, i.e., $(\widetilde{x}_0, \widetilde{y}_0)$, given by the MLE. As sensors are repositioned in each profiling iteration, the discrepancy between the true and estimated profiles is expected to be reduced along with the iterations. The profiling accuracy ω in Eq. (7) is still too complex for us to find efficient movement scheduling algorithms. Hence, we derive the approximation to Eq. (7). If sensors are randomly distributed around the diffusion source, ϵ is close to zero. By assuming a random sensor distribution and setting $\epsilon = 0$, the profiling accuracy can be approximated as:

$$\omega \approx \sum_{i=1}^{N} \omega_i, \quad \omega_i = \sigma^{-2} \mathrm{e}^{-2\beta d_i^2} \left(d_i^2 - \min_{j \in [1,N], j \neq i} d_j^2 \right), \quad (8)$$

where ω_i can be regarded as the *contribution* of sensor i to the overall profiling accuracy. As ω_i depends on d_i and the minimum distance to the source from other sensors, Eq. (8) highly reduces the inter-node dependence compared with Eq. (7). The accuracy of this approximation is validated by extensive numerical results, which are omitted here and available in [30]. In Section 9.3.5, we will evaluate the impact of sensor deployment on the profiling accuracy when the sensor deployment deviates from random distribution around the diffusion source.

6. DIFFUSION PROCESS PROFILING USING ROBOTIC SENSORS

In this section, we formally formulate the movement scheduling problem. Because of the limited mobility and energy budget of aquatic mobile sensors, the sensor movements must be efficiently scheduled in order to achieve the maximum profiling accuracy. As the power consumption of sensing, computation and radio transmission is significantly less than that of locomotion [26], in this paper we only consider the locomotion energy. Moreover, as the locomotion energy is approximately proportional to the moving distance [3], we will use moving distance to quantify the locomotion energy consumption. To simplify the motion control of sensors, we assume that a sensor moves straight in each profiling iteration and the moving distance is always multiple of l meters, where l is referred to as *step*. We note that this model is motivated by the locomotion and computation limitation typically seen for aquatic sensor platforms. First, the locomotion of robotic fish is typically driven by closed-loop motion control algorithms, resulting in constant course-correction

during movement. Second, each profiling iteration has short time duration. As a result, the assumption of sensor's straight movement in an iteration does not introduce significant errors in the movement scheduling. As the estimation and movement are performed in an iterative manner, we will focus on the movement scheduling in one profiling iteration. Denote $m_i \in \mathbb{Z}^+$ and $\phi_i \in [0, 2\pi)$ as the number of steps and movement orientation of sensor i in a profiling iteration, respectively. Our objective is to maximize the expected profiling accuracy after sensor movements, subject to the constraints on total energy budget and sensor's individual energy budget. The movement scheduling problem for diffusion profiling is formally formulated as follows:

Movement Scheduling Problem. *Suppose that a total of M steps can be allocated to sensors and sensor i can move at most L_i meters in a profiling iteration. Find the allocation of steps and movement orientations for all N sensors, i.e., $\{m_i, \phi_i | i \in [1, N]\}$, such that the profiling accuracy ω (defined by Eq. (7)) after sensor movements is maximized, subject to:*

$$\sum_{i=1}^{N} m_i \leq M, \tag{9}$$

$$m_i \cdot l \leq L_i, \quad \forall i. \tag{10}$$

Eq. (9) upper-bounds the total locomotion energy in a profiling iteration. Eq. (10) can be used to constrain the energy consumption of individual sensors. For instance, L_i can be specified according to the sensor's residual energy. Moreover, L_i can also be specified to ensure the delay of a profiling iteration. If sensors move at a constant speed of v m/s and a profiling iteration is required to be completed within τ seconds to achieve the desired temporal resolution of profiling, L_i can be set to $L_i = v \cdot \tau$. As discussed in Section 4, the cluster head adopts MLE to estimate Θ, and then schedules the movements of sensors such that the expected ω in the next profiling iteration is maximized, subject to the constraints in Eqs. (9) and (10). An exhaustive search to the above problem would yield an exponential complexity with respect to N, which is $O((\frac{2\pi}{\phi_0} \cdot \frac{L_i}{l})^N)$ where ϕ_0 is the granularity in searching for the movement orientation. Such a complexity is prohibitively high as the problem needs to be solved in each profiling iteration by the cluster head. In the next section, we will propose an efficient *greedy* algorithm and a near-optimal *radial* algorithm that are feasible to mote-class robotic sensor platform.

7. SENSOR MOVEMENT SCHEDULING ALGORITHMS

In this section, we propose an efficient *greedy* movement scheduling algorithm based on gradient ascent and a near-optimal *radial* algorithm based on dynamic programming to solve the problem formulated in Section 6.

7.1 Greedy Movement Scheduling

Gradient ascent is a widely adopted approach to find a local maximum of a utility function. In this paper, we propose a *greedy* movement scheduling algorithm based on the gradient ascent approach. We first discuss how to determine the movement orientations for the sensors. Since the profiling accuracy ω given by Eq. (7) is a function of all sensors' positions, we can compute the gradient of ω with respect to the position of sensor i (denoted by $\nabla_i \omega$), which is formally given by $\nabla_i \omega = \left[\frac{\partial \omega}{\partial x_i}, \frac{\partial \omega}{\partial y_i}\right]^{\mathrm{T}}$. When all sensors except sensor i remain stationary, the metric ω will increase the fastest if sensor i moves in the orientation given by $\nabla_i \omega$. Therefore, in the greedy movement scheduling algorithm, we let $\phi_i = \angle(\nabla_i \omega)$. Note that sensors will move simultaneously when the movement schedule is executed. We now discuss how to allocate the movement steps. The magnitude of $\nabla_i \omega$, denoted by $\|\nabla_i \omega\|$, quantifies the steepness of the metric ω when sensor i moves in the orientation $\angle(\nabla_i \omega)$ while other sensors remain stationary. Therefore, in the *greedy* algorithm, we propose to proportionally allocate the movement steps according to sensor's gradient magnitude. Specifically, m_i is given by $m_i = \min\left\{\left\lfloor \frac{\|\nabla_i \omega\|}{\sum_{i=1}^{N} \|\nabla_i \omega\|} \cdot M \right\rfloor, \left\lfloor \frac{L_i}{l} \right\rfloor\right\}$. Note that the $\left\lfloor \frac{L_i}{l} \right\rfloor$ in the *min* operator satisfies the constraint Eq. (10). This *greedy* algorithm has linear complexity, i.e., $O(N)$, which is preferable for the cluster head with limited computational resource.

7.2 Radial Movement Scheduling

In this section, we propose a new movement scheduling algorithm based on the approximations discussed in Section 5.3. In this algorithm, each sensor moves toward or away from the estimated source location along the straight line connecting the estimated source location and the sensor's current position. Hence, it is referred to as the *radial* algorithm. We first discuss how to determine sensors' movement orientations and then present a dynamic-programming-based algorithm for allocating movement steps.

From Eq. (8), the contribution of sensor i, ω_i, depends on the minimum distance between the cluster head and other sensors. Because of such inter-node dependence, it is difficult to derive the optimal distance for each sensor that maximizes the overall profiling accuracy ω. It can be shown that the problem involves non-linear and non-convex constrained optimization. Several stochastic search algorithms, such as simulated annealing, can find near-optimal solutions. However, these algorithms often have prohibitively high complexities. In our algorithm, we fix the sensor closest to the estimated source location and only schedule the movements of other sensors in each profiling iteration. As the sensor closest to the source receives the highest SNR, moving other sensors will likely yield more performance gain. Moreover, this sensor can serve as the cluster head that receives measurements from other sensors and computes the movement schedule. It is hence desirable to keep it stationary due to its higher energy consumption in computation and communication. We note that the sensor closest to the source may be different in each iteration after sensor movements, resulting in rotation of cluster head among sensors. By fixing the sensor closest to the source, the distance d_i that maximizes the expected ω_i, denoted by d_i^*, can be directly calculated by

$$d_i^* = \sqrt{\frac{1}{2\beta} + \min_{j \in [1,N]} d_j^2}, \quad \forall i \neq \arg\min_{j \in [1,N]} d_j. \tag{11}$$

Note that as β is a time-dependent variable, d_i^* also changes with time and hence should be updated in each profiling iteration. Eq. (11) allows us to easily determine the movement orientation of sensor i. Specifically, if $d_i > d_i^*$, sensor i will move toward the estimated source location; Otherwise,

sensor i will move in the opposite direction. Formally, by defining $\delta = \text{sgn}(d_i^* - d_i)$, we can express the movement orientation of sensor i as $\phi_i = \angle([\delta \cdot x_i, \delta \cdot y_i]^{\mathrm{T}})$.

We now discuss how to allocate the movement steps. In the rest of this section, when we refer to sensor i, we assume sensor i is not the closest to the estimated source location. After sensor i moves m_i steps in the orientation of ϕ_i, its contribution to the overall profiling accuracy is given by

$$\omega_i(m_i) = \frac{(d_i + \delta \cdot m_i \cdot l)^2 - \min_{j\in[1,N]} d_j^2}{\sigma^2 \cdot \mathrm{e}^{2\beta(d_i+\delta\cdot m_i\cdot l)^2}}, \qquad (12)$$

where $\min_{j\in[1,N]} d_j^2$ in Eq. (12) is a constant for sensor i, and β can be predicted based on its current estimate to capture the temporal evolution of the diffusion, i.e., $\beta = (1/\widetilde{\beta} + 4 \cdot D \cdot \tau)^{-1}$. Given the *radial* movement orientations described earlier, the formulated problem is equivalent to maximizing $\sum_i \omega_i(m_i)$ subject to the constraints Eqs. (9) and (10), which can be solved by a dynamic programming algorithm as follows.

We number the sensors by $1, 2, \ldots, N-1$, excluding the sensor closest to the estimated source location. Let $\Omega(i, m)$ be the maximum ω when the first i sensors are allocated with m steps. Therefore, the dynamic programming recursion that computes $\Omega(i, m)$ can be expressed as:

$$\Omega(i, m) = \max_{0\le m_i\le \lfloor L_i/l\rfloor} \{\Omega(i-1, m-m_i) + \omega_i(m_i)\}.$$

The initial condition of the above recursion is $\Omega(0, m) = 0$ for $m \in [0, M]$. According to the above equation, at the i^{th} iteration of the recursion, the optimal value of $\Omega(i, m)$ is computed as the maximum value of $\lfloor L_i/l \rfloor$ cases which have been computed in previous iterations of the recursion. Specifically, for the case where sensor i moves m_i steps, the maximum profiling accuracy ω of the first i sensors allocated with m steps can be computed as $\Omega(i-1, m-m_i) + \omega_i(m_i)$, where $\Omega(i-1, m-m_i)$ is the maximum ω of the first $i-1$ sensors allocated with $m - m_i$ steps. The maximum overall profiling accuracy is given by $\omega^* = \max_{m\in[1,M]} \Omega(N-1, m)$.

We now describe how to construct the movement schedule using the above dynamic programming recursion. The movement schedule of sensor i is represented by a pair (i, m_i). For each $\Omega(i, m)$, we define a movement schedule $S(i, m)$ initialized to be an empty set. The set $S(i, m)$ is filled incrementally in each iteration when $\Omega(i, m)$ is computed. Specifically, in the i^{th} iteration of the recursion, if $\Omega(i-1, m-m_x) + \omega_i(m_x)$ gives the maximum value among all cases, we add a movement schedule (i, m_x) to $S(i, m)$. Formally, $S(i, m) = S(i-1, m-m_x) \cup \{(i, m_x)\}$, where

$$m_x = \arg\max_{0\le m_i\le \lfloor v\tau/l\rfloor} \{\Omega(i-1, m-m_i) + \omega_i(m_i)\}.$$

The complexity of the dynamic programming is $O\left((N-1)M^2\right)$, where N is the number of sensors and M is the number of allocatable movement steps in a profiling iteration.

8. DISCUSSIONS

8.1 Impact of Localization and Control Errors

The diffusion profiling process discussed in this paper suffers from localization and control errors introduced by GPS module and robotic fish movement. However, our iterative profiling algorithm can largely avoid the accumulation of such errors. First, to achieve desired temporal resolution of profiling, as discussed in Section 6, we upper bound sensors' moving distances in a profiling iteration. Therefore, the cluster head only schedules short-distance movements for sensors in each iteration, hence avoiding the accumulated error in movement control. Moreover, the profiling algorithm avoids the accumulation of localization errors by having sensors update their positions in each profiling iteration. Therefore, the cluster head always leverages the latest sensors' positions that are corrupted only by the errors of current localization. As a result, our profiling algorithm is robust to the localization and control errors. In Section 9.3, we will evaluate the impact of such errors on profiling accuracy using real data traces of GPS and robotic fish movement.

8.2 Scalability of the Radial Algorithm

The complexity of the dynamic programming algorithm presented in Section 7.2 is $O\left((N-1)M^2\right)$. When the number of sensors increases, it is desirable to increase the number of allocatable movement steps. As the maximum moving distance of a sensor is limited by its energy budget, M is often a linear function of N, i.e., $M \sim O(N)$. As a result, the complexity will be $O(N^3)$. Such a complexity may lead to long computation delay at the cluster head, which jeopardizes the timeliness of the periodical profiling process. A basic idea to reduce the computation delay is to bound the number of sensors in each cluster. Although many clustering algorithms can achieve this objective [11], we adopt a simple clustering method, in which each node randomly assigns itself a cluster ID ranging from 1 to p, where p is the total number of clusters. When a diffusion process is profiled by multiple clusters, the dynamic programming procedures are executed separately in different clusters. To account for the interdependence among clusters, the overall estimated profile can be calculated as the average of results from all clusters. In Section 9.2, we will evaluate the trade-off between execution time and profiling accuracy of this algorithm.

9. PERFORMANCE EVALUATION

9.1 Evaluation Methodology

We evaluate our approach through a combination of real testbed experiments and trace-driven simulations. First, we implement the MLE, *greedy* and *radial* algorithms on TelosB motes and evaluate their overhead. The results provide insights into the feasibility of adopting advanced estimation and movement scheduling algorithms on mote-class robotic sensor platforms. Second, we validate the diffusion process model in Eq. (2) with real lab experiments of Rhodamine-B diffusion. Third, we evaluate the proposed profiling algorithm in extensive simulations based on real data traces. We collect three sets of traces, including GPS localization, robotic fish movement control, and on-water Zigbee wireless communication. We analyze the impact of several important factors on the profiling accuracy, including the temporal sampling scheme, the source location bias, the sensor density, and the network communication overhead. Our results show that our approach can accurately profile dynamic diffusion process with low communication overhead.

9.2 Overhead on Sensor Hardware

We have implemented the MLE and the two movement scheduling algorithms in TinyOS 2.1.1 on TelosB platform [15] equipped with an 8 MHz processor. We ported the C

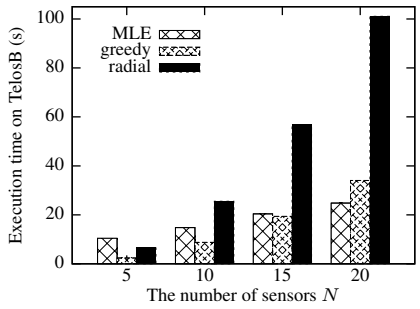


Figure 3: Execution time on TelosB versus the number of sensors N.

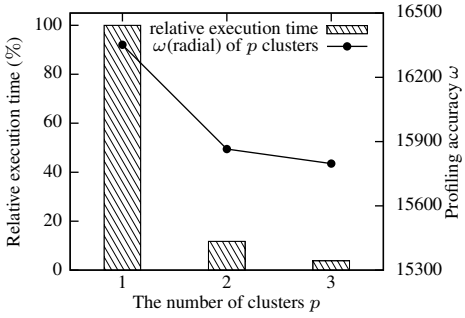


Figure 4: Execution time and profiling accuracy versus the number of clusters p.

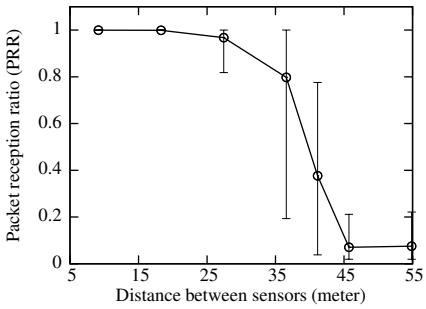


Figure 5: Packet reception ratio (PRR) (90% confidence) versus distance.

implementation of the Nelder-Mead algorithm [18] in GNU Scientific Library (GSL) [2] to TinyOS to solve the optimization problem in MLE (see Section 5.1). The porting is non-trivial because dynamic memory allocation and function pointer are extensively used in GSL while these features are not available in TinyOS. Our implementation of MLE requires 19 KB ROM and 1 KB RAM. When 10 sensors are to be scheduled, the two movement scheduling algorithms require 1 and 8.8 KB RAM, respectively. Fig. 3 plots the average execution time of the MLE, *greedy* and *radial* algorithms versus the number of sensors. We note that the complexity of MLE is $O(N)$. For both movement scheduling algorithms, the execution time linearly increases with N, which is consistent with our complexity analysis. The *radial* algorithm takes about 100 seconds to compute the movement schedule in a profiling iteration when $N = 20$. This overhead is reasonable compared with the movement delay of low-speed mobile sensors. The *greedy* algorithm is significantly faster, and hence provides an efficient solution when the timeliness is more important than profiling accuracy. For the *radial* algorithm, 30% execution time is spent on computing a look-up table consisting of each sensor i's contribution, i.e., $\omega_i(m_i)$ in Eq. (12), given all possible values of m_i. There are several ways to further reduce the computational overhead. First, our current implementation employs extensive floating-point computation. Our previous experience shows that fixed-point arithmetic is significantly more efficient on TelosB motes. Moreover, we can also adopt more powerful sensor platforms as cluster head in the network. For instance, the projected execution time on Imote2 [15] equipped with a 416 MHz processor is within 2 seconds for computing the movement schedule for 20 sensors.

Fig. 4 plots the execution time and the profiling accuracy of the scalable variant of *radial* approach, which is discussed in Section 8.2, versus the number of clusters, respectively. The left Y-axis is the ratio of execution time for p clusters with respect to the case of a single cluster. We can see that both the execution time and profiling accuracy decrease with the number of clusters. As discussed in Section 8.2, this is due to the fact that simply averaging results from all clusters does not fully account for the inter-cluster dependence in the accuracy of dynamic programming. Nevertheless, the *radial* algorithm of 2 and 3 clusters still outperforms the *greedy* algorithm of a single cluster in terms of the profiling accuracy.

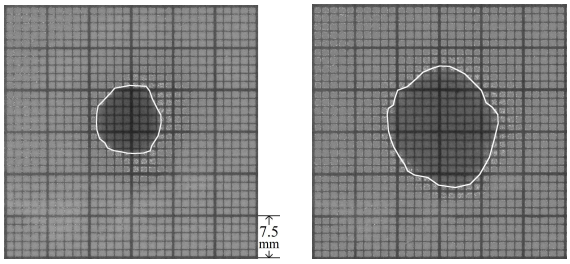
9.3 Trace-Driven Simulations

9.3.1 Model Validation and Trace Collection

We collected four sets of data traces, which include chemical diffusion, GPS localization errors, robotic fish movement control, and on-water Zigbee wireless communication. First, we use the chemical diffusion traces to validate the diffusion process model in Eq. (2). To collect the traces, we discharge Rhodamine-B solution in saline water, and periodically capture diffusion process using a digital camera. We assume that the grayscale of a pixel in the captured image linearly increases with the concentration at the corresponding physical location [27]. Therefore, the evolution of diffusion process can be characterized by the expansion of a contour given a certain threshold of grayscale in the captured images. With the contour areas along the recorded shooting times, we can estimate D by linear regression. The detailed derivations are available in [30]. Fig. 6 plots the captured images with contours marked in white. Fig. 7 plots the contour areas observed in images and predicted by Eq. (2) versus t. We can see that the model in Eq. (2) well characterizes the diffusion of Rhodamine-B.

To evaluate the proposed profiling algorithms, we also collect traces of GPS localization errors, robotic fish movement control and on-water Zigbee wireless communication. First, the data traces of GPS error are collected using two Linx GPS modules [12] in outdoor open space. We extract the GPS error by comparing the distance measured by GPS modules with the groundtruth distance. The average GPS error is 2.29 meters. Second, the data traces of movement control are collected with a robotic fish developed in our lab [26] (see Fig. 1). The movement of robotic fish is driven by a servo motor that is controlled by continuous pulse-width modulation waves. By setting the fish tail beating amplitude and frequency to 23° and 0.9 Hz, the movement speed is 2.5 m/min. We then have the fish swim along a fixed direction in an experimental water tank, and derive the real speed by dividing the moving distance by elapsed time. Third, the data traces of Zigbee communication are collected with two IRIS motes[2] by measuring the packet reception rate (PRR) on the wavy water surface of Lake Lansing on a windy day. We note that the PRR in such wavy water environment is more dynamic than that in calm water environment, due

[2]The next generation of our robotic fish platform adopts the same RF230 radio chip equipped on IRIS.



(a) Elapsed time $t = 3.5\,\mathrm{s}$ (b) Elapsed time $t = 18.1\,\mathrm{s}$

Figure 6: Observations of the diffusion process of Rhodamine-B solution in saline water.

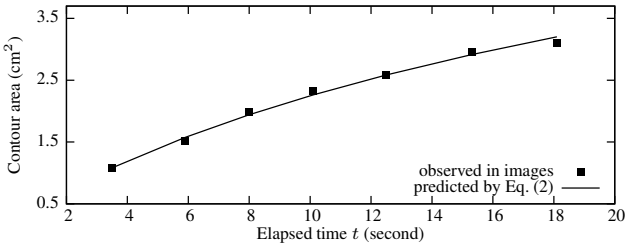
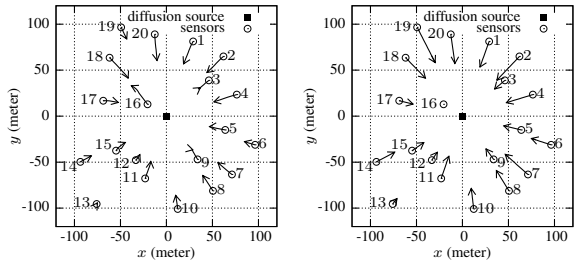


Figure 7: Observed and predicted contour area of diffusing Rhodamine-B in saline water versus elapsed time t.

to multipathing [5]. Specifically, we place the two motes about 12 centimeters above the water surface, and measure the PRR versus the distance between sensors. The results are plotted in Fig. 5. We note that the two IRIS motes achieve an average PRR of 0.8, when they are 37 meters apart. According to our experience, the communication range of IRIS on water surface decreases by about 50% compared to that on land.

9.3.2 Simulation Settings

We conduct extensive simulations based on collected data traces to evaluate the effectiveness of our approach. The simulation programs are written in Matlab. As discussed in Section 3.2, the effect of constant-speed advection is canceled because the sensors and source location are in the same inertial system. Therefore, we only simulate the diffusion process without advection. The diffusion source is at the origin of the coordinate system, i.e., $x_0 = y_0 = 0$. The sensors are randomly deployed in the square region of $200 \times 200\,\mathrm{m}^2$ centered at the origin. The reading of a sensor is set to be the sum of the concentration calculated from Eq. (2), the bias b_i, and a random number sampled from the normal distribution $\mathcal{N}(0, \varsigma^2)$. As discussed in Section 3.2, in each profiling iteration, a sensor samples K readings and outputs the average of them as the measurement. The amount of discharged substance is set to be $A = 0.7 \times 10^6\,\mathrm{cm}^3$ (i.e., $0.7\,\mathrm{m}^3$) unless otherwise specified. The diffusion coefficient is set to be $D = 5{,}000\,\mathrm{cm}^2/\mathrm{s}$. Note that the settings of A and D are comparable to the real field experiments reported in [8] where 2 to $5\,\mathrm{m}^3$ of diesel oil were discharged into the sea and the estimated diffusion coefficient ranged from $2{,}000\,\mathrm{cm}^2/\mathrm{s}$ to $7{,}000\,\mathrm{cm}^2/\mathrm{s}$. The noise standard deviation is set to be $\varsigma = 1\,\mathrm{cm}^3/\mathrm{m}^2$, i.e., $1\,\mathrm{cm}^3$ discharged substance per



(a) *greedy*: final $\omega = 15400$ (b) *radial*: final $\omega = 16400$

Figure 8: Movement trajectories of 20 sensors in the first 15 profiling iterations.

unit area.[3] To easily compare various movement scheduling algorithms, we let the first profiling iteration always start at $t = 1800\,\mathrm{s}$, i.e., half an hour after the discharge. At $t = 1800\,\mathrm{s}$, the average received SNR is around 10/1. The rationale of this setting is that moving sensors too early (i.e., at low SNRs) leads to little improvement on profiling accuracy, resulting in waste of energy. In practice, various approaches can be applied to initiate the profiling process, e.g., by comparing the average measurement to a threshold that ensures good SNRs. Other settings include $l = 0.5\,\mathrm{m}$, $\tau = 60\,\mathrm{s}$, $v = 2.5\,\mathrm{m/min}$ and $K = 2$, unless otherwise specified.

We compare our approach with two additional baseline algorithms in the evaluation. The first baseline (referred to as *SNR-based*) schedules the movements based on the SNRs received by sensors. The SNR received by sensor i (denoted by SNR_i) is defined as $c(d_i, t)/\sigma$, where d_i and t can be computed from the estimated profile $\widetilde{\Theta}$. In the *SNR-based* scheduling algorithm, the sensors always move toward the estimated source location to increase the received SNRs. The movement steps are proportionally allocated according to sensors' SNRs. The rationale behind this heuristic is that the accuracy of MLE increases with SNR. The second baseline (referred to as *annealing*) is based on the simulated annealing algorithm. Specifically, for given movement orientations $\{\phi_i | \forall i\}$, it uses the brutal-force search to find the optimal step allocation under the constraints in Eqs. (9) and (10). It then employs a simulated annealing algorithm to search for the optimal movement orientations. However, it has exponential complexity with respect to the number of sensors.

9.3.3 Sensor Movement Trajectories

We first visually compare the sensor movement trajectories computed by the *greedy* and *radial* movement scheduling algorithms. A total of 20 sensors are deployed. Fig. 8 shows the movement trajectories of sensors in the first 15 profiling iterations. For a particular sensor, the circle denotes its initial position, the segments represent its movement trajectory of 15 profiling iterations, and the arrow indicates its

[3] As we adopt a 2-dimensional model to characterize the diffusion process, the physical unit of concentration is $\mathrm{cm}^3/\mathrm{m}^2$. As observed in the field experiments [8], diesel oil can penetrate down to several meters from the water surface. As a result, the equivalent ς that accounts for the depth dimension ranges from $0.1\,\mathrm{cm}^3/\mathrm{m}^3$ to $1\,\mathrm{cm}^3/\mathrm{m}^3$. Our setting is consistent with the noise standard derivation of the crude oil sensor Cyclops-7 [28], which is $0.1\,\mathrm{cm}^3/\mathrm{m}^3$.

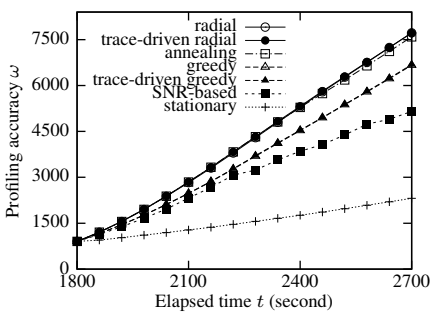


Figure 9: Profiling accuracy ω versus elapsed time t.

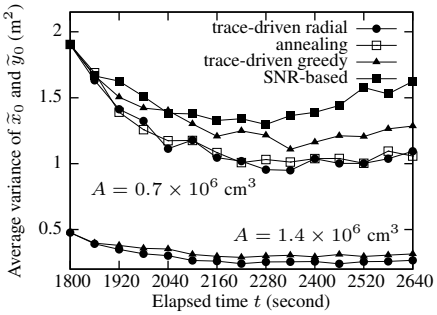


Figure 10: Average $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$ versus elapsed time t.

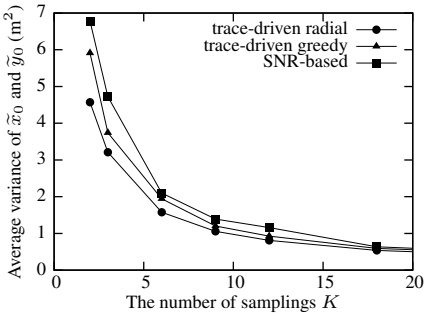


Figure 11: Average $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$ versus the number of samplings K.

movement orientation in the 15th iteration. The sensor with no segments remains stationary during all 15 profiling iterations. We can see that, with the *greedy* algorithm, several sensors (e.g., sensor 15, 16, and 17) have bent trajectories. This is because the movement orientation of each sensor is to maximize the gradient ascent of ω, hence not necessarily to be aligned along the iterations. In the *radial* algorithm, sensors' trajectories are more straight. This is because the movement orientation is along the direction determined by the current sensor position and the estimated source location that is close to the true source location. Moreover, we find that the *radial* algorithm outperforms the *greedy* algorithm in terms of profiling accuracy after the first 15 iterations. In the *greedy* algorithm, the orientation assignment and movement step allocation are based on the gradient derived from the current positions of sensors. Besides, the *greedy* algorithm does not account for the interdependence of sensors in providing the overall profiling accuracy. As a result, its solution may not lead to the maximum ω after the sensor movements and the temporal evolution of diffusion.

9.3.4 Profiling Accuracy

In the second set of simulations, we evaluate the accuracy in estimating the diffusion profile Θ. Total 10 sensors are deployed and our evaluation lasts for 15 profiling iterations. Fig. 9 plots the profiling accuracy ω (defined in Eq. (7)) based on the estimated diffusion profile $\widetilde{\Theta}$. The curve labeled with "stationary" is the result if all sensors always remain stationary. Nevertheless, we can see that the profiling accuracy improves over time because of the temporal evolution of the diffusion. In Fig. 9, the curves labeled with a prefix "trace-driven" are the simulation results based on the data traces of localization errors and robotic fish movement. Specifically, the position reading of a robotic sensor is corrupted by a localization error randomly selected from the GPS error traces. When a robotic sensor moves, its speed is set to be a real speed that is randomly selected from the data traces. We can see that, for both *greedy* and *radial*, the curves with and without simulated movement control and localization errors almost overlap with each other. As our iterative approach has no accumulated error, small movement control and localization errors have little impact on our approach. The *radial* algorithm outperforms the *greedy* and *SNR-based* algorithms by 16% and 50% in terms of ω at the 15th profiling iteration, respectively. And the accuracy performance of the *radial* algorithm is very close to the *annealing* algorithm that can find the near-optimal solution. However, we note that in each iteration of the *annealing* algorithm, a new look-up table needs to be computed due to changed movement orientations. Hence its execution time highly depends on the number of iterations that can be very large. Therefore, the *annealing* algorithm is infeasible on mote-class platforms.

Fig. 10 plots the average of $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$ in each profiling iteration under various settings of the discharged substance amount A. In order to evaluate the variances in each profiling iteration, the sensors perform many rounds of MLE, where each round yields a pair of $(\widetilde{x}_0, \widetilde{y}_0)$. The $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$ are calculated from all rounds. From Fig. 10, we find that the variances may increase (for the *SNR-based* scheduling algorithm) or fluctuate (for other approaches) after several iterations. This is because the variances are time-dependent due to the involving of α and β in $\mathrm{CRB}(x_0)$ and $\mathrm{CRB}(y_0)$. Moreover, we can see that the variances decrease with A. As sensors receive higher SNRs in the case of higher A, our result consists with the intuition that the estimation error decreases with SNR. Compared with the *SNR-based* algorithm, the *radial* algorithm reduces the variance in estimating diffusion source location by 36% for $A = 0.7 \times 10^6\,\mathrm{cm}^3$. Compared with the *greedy* algorithm, the reductions are 12% and 18% for $A = 0.7 \times 10^6$ and $1.4 \times 10^6\,\mathrm{cm}^3$, respectively. We also evaluate the accuracy in estimating the substance amount A and elapsed time t. Both the *greedy* and *radial* algorithms can achieve a high accuracy. For instance, the relative error in estimating A for *radial* algorithm is within 1.4%. Due to space limitation, detailed evaluation results are omitted here and can be found in [30].

9.3.5 Impact of Sampling, Source Bias and Network Density

We characterize the profiling error after 15 iterations by the average of $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$. Except for the evaluations on network density, a total of 10 sensors are deployed. In the temporal sampling scheme presented in Section 3.2, a sensor yields the average of K continuous samples as the measurement to reduce noise variance. Fig. 11 plots the profiling error versus K. We can see that the profiling error decreases with K. The relative reductions of profiling error by the *radial* algorithm with respect to the *greedy* and *SNR-based* algorithms are about 18% and 30%, respectively, when K ranges from 2 to 20.

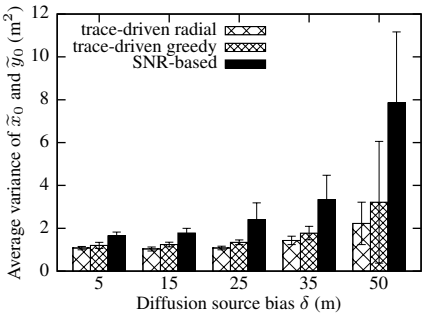


Figure 12: Average $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$ versus source location bias δ.

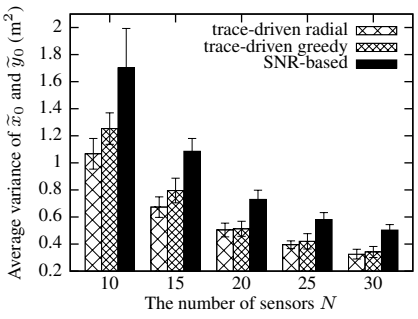


Figure 13: Average $\mathrm{Var}(\widetilde{x}_0)$ and $\mathrm{Var}(\widetilde{y}_0)$ versus the number of sensors N.

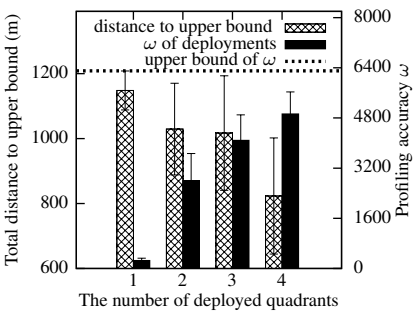


Figure 14: Impact of initial deployment on the profiling accuracy.

The approximations discussed in Section 5.3 assume that the sensors are randomly deployed around the diffusion source. In this set of simulations, we evaluate the impact of source location bias on profiling accuracy. Specifically, the diffusion source appears at $(\delta, 0)$, where δ is referred to as the source location bias. Fig. 12 plots the profiling error versus δ. To jointly account for the impact of random sensor deployment, for each setting of δ, we deploy a number of networks and show the error bars. We find that the *radial* algorithm is robust to the source location bias. Moreover, we note that the *radial* algorithm is consistently better than other algorithms.

Fig. 13 plots the profiling error versus the number of sensors. When more sensors are deployed, the profiling error can be reduced. The *radial* algorithm is consistently better than the other algorithms. For all algorithms, the profiling error is reduced by about 40% when the number of sensors increases from 10 to 15. Moreover, the relative reduction of profiling error decreases with the number of sensors.

9.3.6 Impact of Sensor Deployment

In this section, we evaluate the impact of initial sensor deployment on the profiling accuracy and energy consumption in locomotion. We fix each d_i and randomly deploy sensors in one, two adjacent, three and four quadrants of the plane originated at the source location, resulting in four sensor deployments. We compute the upper bound of ω, in which sensors' angles with respect to the source location are exhaustively searched to maximize the profiling accuracy. Note that the sensor deployment with maximized profiling accuracy is still an open issue. Fig. 14 plots the upper bound of ω as well as the profiling accuracy of four sensor deployments. We can see that the profiling accuracy of the four-quadrant deployment is the closest to the upper bound. Fig. 14 also plots the minimum total distance that the sensors in a deployment have to move to achieve the upper bound ω. We can observe that if sensors are not deployed around the source location, spreading sensors first can significantly improve the profiling accuracy. However, if sensors have limited energy for locomotion, it is more beneficial to deploy sensors around the source to avoid energy-consuming spreading movements.

9.3.7 Communication Overhead

We conduct a set of trace-driven simulations to evaluate the communication overhead of our approach. Specifically, we choose the shortest distance path as the routing path from a sensor to the cluster head, where the distance metric of each hop is PRR^{-1}, i.e., the expected number of (re-)transmissions on the hop. When a node transmits packet to the next hop, the packet is delivered with a success probability equal to the PRR retrieved from the communication traces with the same distance between the sender and receiver. The node re-transmits the packet for 10 times before it is dropped until success. In the simulations, 30 sensors are randomly deployed. The packet to the cluster head includes sensor ID, current position and measurement, and the packet to the sensor includes moving orientation and distance. Our simulation results show that the number of packet (re-)transmissions in a profiling iteration has a mean of 158 and a standard deviation of 28. Even if all these transmissions happen sequentially, the delay will be within seconds, because transmitting a TinyOS packet only takes about 10 milliseconds. This result shows that our approach has low communication overhead under realistic settings.

10. CONCLUSION AND FUTURE WORK

In this paper we propose an accuracy-aware profiling approach for aquatic diffusion processes using robotic sensor networks. Our approach features an iterative profiling process where the sensors reposition themselves to progressively improve the profiling accuracy along the iterations. We develop two movement scheduling algorithms, including an efficient *greedy* algorithm and a near-optimal *radial* algorithm. We implement our algorithms on TelosB motes and evaluate their overhead. We also conduct extensive simulations based on real traces of chemical diffusion, GPS localization errors, robotic fish movement, and wireless communication. Our results show that our approach can accurately profile dynamic diffusion processes with low overhead.

The movement scheduling approach described in this paper is targeted at robotic sensors with limited sensing and motion capabilities in relatively calm water environment. We are developing the next generation of our robotic fish platforms that are capable of more complex sensing and motion control. In our future work, we will investigate distributed control algorithms that allow such robotic sensors to autonomously plan their motion paths, which reduce the overhead of cross-sensor coordination in collaborative sensing tasks. In addition, we will extend our approach to address wavy water environment by quantifying the impact of waves on sensor measurement and wireless link quality.

Acknowledgment

This research was supported in part by the U.S. National Science Foundation under grants ECCS-1029683 and CNS-0954039 (CAREER). We thank Ruogu Zhou for his contribution in traces collection and hardware design. We also thank the anonymous reviewers providing the valuable feedbacks.

11. REFERENCES

[1] http://www.infoplease.com/ipa/A0001451.html.

[2] GSL-GNU scientific library, 2011.

[3] D. Barrett, M. Triantafyllou, D. Yue, M. Grosenbaugh, and M. Wolfgang. Drag reduction in fish-like locomotion. *Journal of Fluid Mechanics*, 392(1):183–212, 1999.

[4] J. Chin, D. Yau, N. Rao, Y. Yang, C. Ma, and M. Shankar. Accurate localization of low-level radioactive source under noise and measurement errors. In *The 6th ACM conference on Embedded Networked Sensor Systems (SenSys)*, pages 183–196, 2008.

[5] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore. Environmental wireless sensor networks. *Proceedings of the IEEE*, 98(11):1903–1917, 2010.

[6] J. Crank. *The mathematics of diffusion*. Oxford University Press, 1983.

[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

[8] A. Elliott. Shear diffusion and the spread of oil in the surface layers of the north sea. *Ocean Dynamics*, 39(3):113–137, 1986.

[9] C. Eriksen, T. Osse, R. Light, T. Wen, T. Lehman, P. Sabin, J. Ballard, and A. Chiodi. Seaglider: A long-range autonomous underwater vehicle for oceanographic research. *IEEE Journal of Oceanic Engineering*, 26(4):424–436, 2001.

[10] Hydroid, LLC. REMUS: Autonomous technology for your world.

[11] V. Kumar. *Introduction to parallel computing*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[12] Linx Technologies. Linx GPS receiver module data guide.

[13] N. March and M. Tosi. *Introduction to Liquid State Physics*. World Scientific Publishing, 2002.

[14] J. Matthes, L. Groll, and H. Keller. Source localization by spatially distributed electronic noses for advection and diffusion. *IEEE Transactions on Signal Processing*, 53(5):1711–1719, 2005.

[15] Memsic Corp. TelosB, IRIS, Imote2 datasheets.

[16] S. Murray. Turbulent diffusion of oil in the ocean. *Limnology and oceanography*, 17(5):651–660, 1972.

[17] A. Nehorai, B. Porat, and E. Paldi. Detection and localization of vapor-emitting sources. *IEEE Transactions on Signal Processing*, 43(1):243–253, 1995.

[18] J. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308, 1965.

[19] L. Rossi, B. Krishnamachari, and C. Kuo. Distributed parameter estimation for monitoring diffusion phenomena using physical models. In *The 1st annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 460–469, 2004.

[20] S. Ruberg, R. Muzzi, S. Brandt, J. Lane, T. Miller, J. Gray, S. Constant, and E. Downing. A wireless internet-based observatory: The real-time coastal observation network (recon). In *OCEANS*, pages 1–6, 2007.

[21] D. Rudnick, R. Davis, C. Eriksen, D. Fratantoni, and M. Perry. Underwater gliders for ocean research. *Marine Technology Society Journal*, 38(2):73–84, 2004.

[22] A. Singh, R. Nowak, and P. Ramanathan. Active learning for adaptive mobile sensing networks. In *The 5th international conference on Information Processing in Sensor Networks (IPSN)*, pages 60–68, 2006.

[23] Z. Song, Y. Chen, J. Liang, and D. Uciński. Optimal mobile sensor motion planning under nonholonomic constraints for parameter estimation of distributed systems. *International Journal of Intelligent Systems Technologies and Applications*, 3:277–295, 2007.

[24] S. Srinivasan, K. Ramamritham, and P. Kulkarni. Ace in the hole: Adaptive contour estimation using collaborating mobile sensors. In *The 7th international conference on Information Processing in Sensor Networks (IPSN)*, pages 147–158, 2008.

[25] R. Tan, G. Xing, J. Wang, and H. So. Exploiting reactive mobility for collaborative target detection in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(3):317–332, 2010.

[26] X. Tan. Autonomous robotic fish as mobile sensor platforms: Challenges and potential solutions. *Marine Technology Society Journal*, 45(4):31–40, 2011.

[27] E. Tsotsas and A. Mujumdar. Modern drying technology, vol. 2, experimental techniques, 2009.

[28] Turner Designs Inc. Cyclops-7 user's manual.

[29] S. Vijayakumaran, Y. Levinbook, and T. Wong. Maximum likelihood localization of a diffusive point source using binary observations. *IEEE Transactions on Signal Processing*, 55(2):665–676, 2007.

[30] Y. Wang, R. Tan, G. Xing, J. Wang, and X. Tan. Accuracy-aware aquatic diffusion process profiling using robotic sensor networks. Technical report, CSE Department, Michigan State University, 2011.

[31] J. Weimer, B. Sinopoli, and B. Krogh. Multiple source detection and localization in advection-diffusion processes using wireless sensor networks. In *The 30th Real-Time Systems Symposium (RTSS)*, pages 333–342, 2009.

[32] G. Xing, J. Wang, Z. Yuan, R. Tan, L. Sun, Q. Huang, X. Jia, and H. So. Mobile scheduling for spatiotemporal detection in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(12):1851–1866, 2010.

[33] T. Zhao and A. Nehorai. Distributed sequential bayesian estimation of a diffusive source in wireless sensor networks. *IEEE Transactions on Signal Processing*, 55(4):1511–1524, 2007.