

Sea Depth Measurement with Restricted Floating Sensors

MO LI, Nanyang Technological University, Singapore
 ZHENG YANG and YUNHAO LIU, Tsinghua University, China

Sea depth monitoring is a critical task for ensuring safe operation of harbors. Traditional schemes largely rely on labor-intensive work and expensive hardware. This study explores the possibility of deploying networked sensors on the surface of the sea, measuring and reporting the sea depth of given areas. We propose a Restricted Floating Sensors (RFS) model in which sensor nodes are anchored to the sea bottom, floating within a restricted area. Distinguished from traditional stationary or mobile sensor networks, the RFS network consists of sensor nodes with restricted mobility. We construct the network model and elaborate the corresponding localization problem. We show that by locating such RFS sensors, the sea depth can be estimated without the help of any extra ranging devices. A prototype system with 25 Telos sensor nodes is deployed to validate this design. We also examine the efficiency and scalability of this design through large-scale simulations.

Categories and Subject Descriptors: C.2.4 [Computer-Communication Networks]: Distributed Systems—*Distributed applications*; C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Distributed networks, Network communications*

General Terms: Algorithms, Design, Measurement

Additional Key Words and Phrases: FALA, sea depth measurement, restricted floating sensors, floating area localization

ACM Reference Format:

Li, M., Yang, Z., and Liu, Y. 2013. Sea depth measurement with restricted floating sensors. *ACM Trans. Embedd. Comput. Syst.* 13, 1, Article 1 (August 2013), 21 pages.
 DOI: <http://dx.doi.org/10.1145/2512448>

1. INTRODUCTION

We conducted a field study in Huanghua Harbor which is currently the second largest harbor for coal transportation in China. It has experienced rapid development over the past five years, and its coal transporting capability has increased from 1.6 million tons per year in 2002 to 6.7 million tons per year in 2006. However, this Harbor currently suffers from the increasingly severe problem of Silt deposition along its sea route. Huanghua Harbor has a sea route that is 19 nautical miles long and 800 m wide at the entrance, including an inner route and an outer route. The sea route is designed to have a water depth of 13.5 m to allow for the passage of ships that weigh over 50,000 tons. Since the sea route has been in operation, it has always been threatened

A preliminary version of this work was published in *Proceedings of the 28th IEEE Real-Time Systems Symposium* [Yang et al. 2007].

This work is supported by Singapore MOE Tier 2 under MOE2012-T2-1-070, National Basic Research 973 Program of China under grant No. 2011CB302705, NSF China Major Program 61190110, and NSF China Distinguished Young Scholars Program 61125202.

Authors' addresses: M. Li (corresponding author), School of Computer Engineering, Nanyang Technological University; email: limo@ntu.edu.sg; Z. Yang and Y. Liu, School of Software and TNLIST, Tsinghua University. Permission to make digital or hard copies of part or all 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 show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2013 ACM 1539-9087/2013/08-ART1 \$15.00

DOI: <http://dx.doi.org/10.1145/2512448>

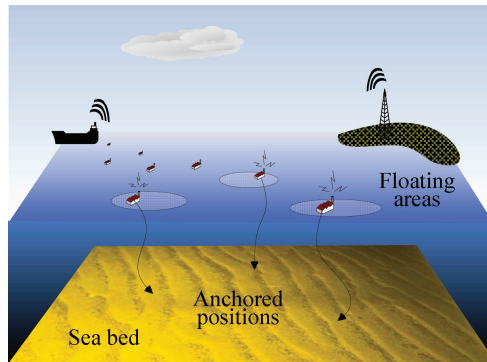


Fig. 1. RFS network on the sea.

by the movement of silt from the shallow sea area within 14 nautical miles outside the route entrance. In the event that the sea route is silted up, ships of large tonnage would have to wait to enter the harbor to prevent grounding, and ships of small tonnage would need be piloted into the harbor. Monitoring the extent of siltation reliably is critical in order to ensure the safe operation of Huanghua Harbor.

The uncertainty and the high-instant intensity of the siltation make monitoring the extent of siltation extremely expensive and difficult. The amount of siltation in Huanghua Harbor is affected by many factors, among which tide and wind blow are the most dominating. While the tides produce a periodic influence on the movement of silt, the highly variable nature of wind brings more incidental and intensive effects. For example, records show that strong winds with wind forces of 9 to 10 on the Beaufort scale hit Huanghua Harbor from October 10 to October 13, 2003. The storm surge brought 970,000 m³ of silt to the sea route, which suddenly decreased the water depth from 9.5 m to 5.7 m and blocked most of the ships weighing more than 35,000 tons. Harbor administration hired three boats equipped with active sonars to cruise the 380 km² shallow sea area around the harbor for several days. Monitoring sea depth costs this harbor more than 18 million U.S. dollars per year.

In this work, we explore the possibility of deploying networked sensors on the surface of the sea for sea depth measurement, which would provide inexpensive alternatives to addressing the problem. We anchor the sensor nodes by ropes and let them float on the sea. In this case, sensor nodes, called *Restricted Floating Sensors* (RFS) float over the sea but are restricted within their anchored areas. The floating area is dependent on the sea level and the length of the rope. Figure 1 illustrates an RFS network deployed in a sea area.

To map the sea depth, an essential problem is localizing RFS nodes. Since the sensor nodes in the RFS network can float around, the traditional localization approaches for stationary sensors provide inaccurate measurements or intolerable latencies. On the other hand, simply treating the RFS network as a mobile sensor network and blindly applying those localization approaches for mobile WSNs does not capture the special nature of the RFS network. In the RFS network, sensor nodes float within restricted areas, providing us possibilities for capturing their mobility models. By understanding RFS mobility behaviors, we can achieve higher accuracy with reduced overhead.

In this article, we give an elaborate analysis on the localization problem in the RFS network. We build network models and establish the localization objective as locating the floating area of each sensor node in the RFS network. Unlike locating the instant locations of each sensor in unlimited mobile networks, RFS sensors are

anchored. Hence, locating the floating areas of sensor nodes is sufficient for obtaining the geographic features of the RFS network. Taking snapshots of mobile sensors does not give the desired result of their floating areas, yet instantly computing the snapshot positions of those mobile sensors is very difficult and costly. We equip a small portion of the network nodes with external locating devices, such as GPS receivers (called seed nodes), and all sensor nodes estimate their distances from each other. We explore different statistical approaches for efficiently localizing the floating areas of the seed and non-seed nodes with distance estimations. By locating the sensor nodes, we can infer accordingly the sea depth at the anchor positions in a practical way, which will be elaborated on in Section 5. Such a method could be applied for other applications where detecting the sea depth is a primary goal. FALA could also be used in other application scenarios where sensor nodes could be of constrained-mobility and where detecting the mobility area of sensors suffices to approximate the locations of sensors. Using FALA in such scenarios would save unnecessary overhead in traditional localization approaches, such as accurate synchronization, fast positioning, unnecessary energy consumption, etc.

We validate our design by launching a prototype system with 25 Telos sensor nodes off the seashore in our campus. The results show that our prototype achieves less than 0.5 m sea depth estimation error on average. We conduct a large-scale simulation to further test the system performance and scalability under various network settings. With precise distance measurements assumed, we can obtain the sea depth estimation with an average relative error within 20%.

The rest of the article is organized as follows. In Section 2, we formally define the RFS network model and formulate the localization problem for RFS. We describe our localization approaches for seeds and non-seeds in Sections 3 and 4, respectively. In Section 5, we discuss applying the RFS localization for sea depth measurement. We present the experiments and results in Section 6. In Section 7, we summarize related work and conclude in Section 8.

2. THE NETWORK MODEL

Before presenting our targeted Restricted Floating Sensor (RFS) network, we first give a definition of the more general restricted mobile sensor (RMS) network.

Definition 2.1 (RMS Network). A sensor is called a restricted mobile sensor if it is capable of movement but its movement is restricted within a local area of the application field. A network composed of restricted mobile sensors is called an RMS network. The RFS is a typical RMS network. Once anchored at a point, the sensor node floats on the sea surface but within a restricted area. There are some other RMS networks, for example, in Li et al. [2008], mobile sensors are restricted to moving along the cables. Such RMS networks differ from the considered RFS network in this article as sensors are restricted within different types of areas.

Definition 2.2 (Floating Area). In an RMS network, the movement of a sensor is limited in a restricted area. The restricted area may have different shapes due to different constraints of RMS networks. In the RFS network, each sensor node floats on the sea surface within a disk area centered at its anchor. This disk area is called the floating area of the sensor. We use $o(c, r)$ to denote a disk floating area, where c and r represent the centre and radius of the disk area, respectively. In practice, c is the anchored position of each sensor, and r is determined by the length of the rope and the sea depth at the anchored position.

Definition 2.3 (Floating Model). In the RFS network, each sensor floats within its floating area. The movement is affected by many factors, for example, ocean current, wind blow, tide, etc. These factors are hard to model and mostly affect with randomness. Sensor nodes could possibly be applied with high velocities within its restricted floating area such that the current position of a sensor is considered independent of its previous positions under nonnegligible intervals between consecutive sampling times. When the rope becomes straight, the node on the sea surface moves to the boundary of the floating area, that is, the farthest to the center of the disk. When the rope is not straight, the node simply randomly appears within the floating area. Each sensor is assumed to appear in the floating area under uniform distribution, and the probability distribution of the sensor position is given as follows.

$$f(x, y) = \begin{cases} \frac{1}{\pi r^2}, & (x, y) \in o(c, r), \\ 0, & \text{otherwise.} \end{cases}$$

Definition 2.4 (RFS Network Model). The targeted RFS network $N(S, O)$ consists of a set of sensors S and the corresponding set of floating areas O . Each sensor s_i moves within its floating area o_i under the floating model. The floating areas of different sensors are assumed non-overlapped, that is, $\forall s_i, s_j \in S$ within floating area $o_i(c_i, r_i)$ and $o_j(c_j, r_j)$, $\text{dist}(c_i, c_j) > r_i + r_j$. Such an assumption prevents the possibility that two sensors get too close and that their ropes get twisted with each other. The assumption is realistic in practice, as the sensor communication range is usually multiple times the radius of the sensor floating area.

Definition 2.5 (Neighborhood of RFS). In traditional sensor networks, the neighbors of a sensor s are defined as the set of sensors that have direct communications with s . While the neighborhood is relatively stable in static sensor networks, it is highly dynamic in mobile sensor networks. As a restricted mobile sensor network, an RFS network shares the similarity with traditional mobile sensor networks in that each sensor node has dynamic connections with its neighboring nodes. However, the locality of sensor movement in the RFS network constrains this dynamic effect. Therefore, we are able to introduce a more proper definition of neighborhood for RFS. Sensors s_i and s_j are defined to be neighbors if and only if they can communicate with each other in their entire floating areas. Each node has direct communications with its neighbor nodes at any time. With this definition, we obtain a stable neighborhood in RFS networks.

Definition 2.6 (Seeds and Non-seeds). Seeds refer to the sensor nodes equipped with localization devices and are aware of their instantaneous locations. Non-seeds refer to those nodes which cannot directly obtain their instantaneous locations. To reduce cost, our proposed RFS network only employs a small portion of seeds.

Definition 2.7 (Floating Area Localization). In RFS networks, sensor nodes move within their floating areas under the probabilistic floating model. Hence, it is difficult to locate their instantaneous locations. We observe that the floating areas do provide geographical location information of sensor nodes. Thus, the localization issue in RFS networks is to obtain the floating areas instead of the instantaneous locations. The floating area localization in the RFS network indicates the process of locating the floating area $o(c, r)$ of each sensor, including the central anchor position c and the radius r of the floating area. In the following, localization means floating area localization if not elsewhere specified.

Definition 2.8 (Error). Let $o(c, r)$ be the floating area of sensor s and $\hat{o}(\hat{c}, \hat{r})$ be the estimated floating area. The localization error includes two parts: (1) error on the

estimated anchor position $e_c = \text{dist}(\hat{c}, c)$; (2) error on the estimated radius $e_r = |\hat{r} - r|$. We define the floating area localization error as a 2D vector (e_c, e_r) . The relative error ε of floating area $o(c, r)$ is accordingly defined as $E(\hat{o}, o) = (e_c/r, e_r/r)$. The average error in a RFS network $N(S, O)$ is defined as

$$E(N) = \frac{1}{|O|} \sum_{o \in O} E(\hat{o}, o).$$

We design the Floating Area Localization Algorithm (FALA) to localize sensor nodes in an RFS network. In the localization process, all sensor nodes are able to measure the distances between themselves through RSS measurements. Other superior techniques like TOA, TDOA, and AOA can be applied for higher-ranging accuracy.

As a statistic-based algorithm, FALA yields the localization result after a series of data sampling. During each sampling process, seeds collect their locations, and non-seeds process the distance measurements. FALA applies different schemes for locating seeds and non-seeds. Although seeds are able to know their instantaneous locations, further computation based on the location information is needed to determine their floating areas. For non-seeds, FALA derives their floating areas from distance information through a sequential process.

FALA includes four steps: sampling, seed floating area computing, non-seed floating area computing, and continuous data collection and accuracy improvement.

3. FALA FOR SEEDS

As equipped with localization devices, seeds are aware of their instant positions. We carry out a series of samplings on seed positions. After a period of time, each seed node records a set of positions it resides in at different times. We estimate the floating area of seed nodes from the position sets.

Obviously, all sampled positions of a seed are certainly in its floating area under the floating model. In other words, its floating area should be a disk area at least containing all sampled positions. Moreover, as the sampled positions accumulate, the floating area is asymptotically approached. Thus, we can transform the floating area localization problem to figure out a disk area which covers a set of positions.

Apparently, there are many feasible disk areas, among which the smallest one should be considered the maximum likelihood estimation because it provides the highest probability of the occurrence of a set of positions. Thus, the smallest one is then considered the estimation of the floating area of the seed. The problem is formulated as follows.

Given a set P of n points in the plane, find the smallest enclosing disk for P , that is, the smallest disk that contains all the points of P .

For simplicity, we assume that no three points are collinear and that no four points are cocircular. In computational geometry, this problem is often called the Minimum Enclosing Disk (MED) problem. It is not difficult to find a brute force solution to the problem which takes $O(n^4)$ running time. We observe that the MED solution must contain at least two points on its boundary; otherwise, one can shrink the disk without losing any points. Also, two or three points define a disk in the plane. In the case of two points, they must define a diameter of the disk. Therefore, the seed can validate the candidate disks by enumerating all $O(n^2)$ pairs and $O(n^3)$ three-tuples of the recorded positions. For each candidate disk, the seed checks in $O(n)$ time whether the disk contains all other points. In all, this algorithm takes $O(n^4)$ time. Such an algorithm introduces intensive computational cost which is likely not suitable for the resource-restricted sensor nodes. We introduce a randomized algorithm *RMED*, which takes only $O(n)$ expected runtime to compute the MED. We further develop an online

version *RMED_ONLINE* based on *RMED* which incrementally updates the input seed positions and takes $O(1)$ expected runtime for the updating process at each sampling time. We discuss the *RMED* algorithm in Section 3.1 and describe *RMED_ONLINE* in Section 3.2. We analyze the accuracy and runtime of *RMED_ONLINE* in Section 3.3.

3.1. A Randomized Algorithm *RMED*

A randomized algorithm [de Berg et al. 2000; Welzl 1991] for the MED problem has been proposed in the computational geometry domain, which takes $O(n)$ expected runtime. It is observed that when a point is outside the MED of all other points, it must lie in the boundary of the MED of all points. The following theorem [de Berg et al. 2000] illustrates this observation.

THEOREM. *Let P be a set of points in the plane. Let R be a possibly empty set of points with $R \cap P = \emptyset$. Let $D(P, R)$ denote the minimum enclosing disk of P that contains R on its boundary. Then we have the following.*

- (a) *If a point $p \in D(P \setminus \{p\}, R)$, then $D(P, R) = D(P \setminus \{p\}, R)$.*
- (b) *Otherwise, $D(P, R) = D(P \setminus \{p\}, R \cup \{p\})$.*

Based on this theorem, the randomized algorithm *RMED* computes the MED of a given set P of positions. At the very beginning, we have no idea about which point lies on the boundary of MED, so the seed runs *RMED*(P, null) as a start.

ALGORITHM 1: *RMED* (P, R)

```

1: if  $P = \emptyset$  or  $|R| = 3$ ,
2:   then  $D :=$  the disc defined by  $R$ .
3: else choose a random  $p \in P$ ,
4:    $D := \text{RMED}(P \setminus \{p\}, R)$ ;
5:   If  $p \notin D$ ,
6:     then  $D := \text{RMED}(P \setminus \{p\}, R \cup \{p\})$ .
7: return  $D$ .
```

In *RMED*, it is not necessary to generate a random point $p \in P$ in each recursive call. It suffices to randomly permute the input points and then examine points in this random order during the recursive calls. That is, we insert a process to generate a random permutation $(p_1, p_2 \dots p_n)$ of P before executing the *RMED* process. *RMED* sequentially selects one p_i in $(p_1, p_2 \dots p_n)$ and treats p_i as the random point p .

Due to the permutation process, a seed needs to collect all sampled positions in P before it is able to start the algorithm, which means a long latency before the area can be localized. To address this issue, we design an online version of *RMED* for stepwise updating of the floating area approximation.

3.2. *RMED_ONLINE* for Stepwise Approximation

When a seed node is deployed, the location samples are collected sequentially at consecutive sampling times. Let $(t_1, t_2 \dots t_n)$ denote a series of sampling times. Let p_i denote seed position at sampling time t_i ($1 \leq i \leq n$). Each sampled p_i is in fact a random position in the n position set $\{p_1, p_2 \dots p_n\}$, though at this stage, the seed has not obtained the intact knowledge of all p_j ($i < j \leq n$). The sequentially sampled series $(p_1, p_2 \dots p_n)$ is indeed a random permutation instance of the n position set.

In *RMED_ONLINE*, the seed maintains and updates the current MED at each sampling stage. Assume a seed already has the MED for the previous $i - 1$ positions. When

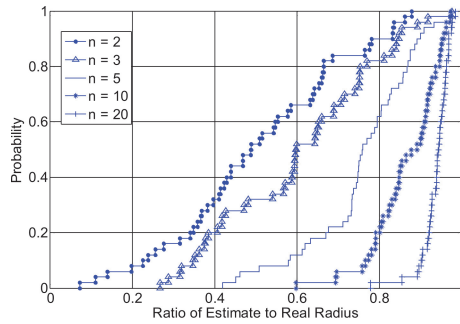


Fig. 2. Empirical cumulative distribution function of the estimated to real radius ratio.

the i th position is obtained, the seed updates the existing MED to obtain a feasible one for all i positions.

Let P_i be the set of points $\{p_1, p_2, \dots, p_i\}$; let D_i be the MED of P_i ; and let p_i be the current position of seed at sampling time t_i . After obtaining p_i , the seed updates its existing solution D_{i-1} by executing a process $D_i = \text{RMED_ONLINE}(p_i, D_{i-1})$. The *RMED_ONLINE* process is described as follows.

ALGORITHM 2: *RMED_ONLINE* (p_i, D_{i-1})

- 1: if $p_i \in D_{i-1}$,
 - 2: then $D := D_{i-1}$;
 - 3: otherwise,
 - 4: $D := \text{RMED}(P_{i-1}, \{p_i\})$;
 - 5: return D .
-

In the *RMED_ONLINE* process, seeds can start localization as early as possible without waiting for all n positions to be collected. This feature of *RMED_ONLINE* well suits the data acquisition pattern of the seed sampling process. Each time, being informed of the current position from its positioning device, a seed updates its existing minimum enclosing disk and obtains a refined approximation of the floating area. The estimation is refined in a stepwise manner while, as shown in next section, each intermediate updating process takes only $O(1)$ expected runtime.

3.3. *RMED_ONLINE* Analysis

In this section, we analyze the error ratio and the runtime of *RMED_ONLINE*. According to our algorithm, the approximated floating area \hat{o} is always smaller than the real one o . The error between o and \hat{o} is kept reduced during the updating processes in which \hat{o} is monotonously increased towards o . To minimize the estimation error, the seed needs to collect more sample data. However, a large sample capacity usually implies a long period of sampling. Therefore, we need to properly choose a sample capacity aiming for an acceptable accuracy.

We conduct a simulation to analyze the error of our estimation at different sample capacities of $n = 2, 3, 5, 10$, and 20 . The simulation results are shown in Figure 2. We find that when the number of samplings is 10, 80% percent of cases have less than 20% relative error of radius estimation, and when the number of samplings is 20, 90% percent of cases have less than 10% error. In most applications, a number ranging from 10 to 20 induces an acceptable sample capacity for seeds to compute their floating areas.

To analyze the runtime of the *RMED* process, we focus on the number of tests $p \notin D$, because the total execution complexity is proportional to this number. Let $T_j(n)$ be the expected number of tests performed in the *RMED* process such that $|P| = n$ and $|R| = 3 - j$, where $0 \leq j \leq 3$. Note that $T_0(n) = 0$ for all n because only lines 1 and 2 of *RMED* algorithm are executed. For $j \geq 1$ and $n \geq 1$, line 4 takes $T_j(n - 1)$ time; line 5 performs the test once; and line 6 may take further $T_{j-1}(n - 1)$ time. In order to take further $T_{j-1}(n - 1)$ time, p must lie on the boundary of the solution disk. Since $|R| = 3 - j$, only j out of the current n points can do so. The probability that p is one of them is j/n . Therefore, we have the following recurrence.

$$T_j(n) \leq T_j(n - 1) + \frac{j}{n} T_{j-1}(n - 1).$$

Since $T_0(n) = 0$, we obtain $T_1(n) \leq T_1(n - 1) + 1$, implying that $T_1(n) \leq n$. Then

$$T_2(n) \leq T_2(n - 1) + 1 + 2(1 - 1/n) < T_2(n - 1) + 3,$$

implying that $T_2(n) \leq 3n$. Then

$$T_3(n) \leq T_3(n - 1) + 1 + 9(1 - 1/n) \leq T_3(n - 1) + 10,$$

implying that $T_3(n) \leq 10n$.

Now, we turn to the asymptotic complexity of the *RMED_ONLINE* process. Similarly, we focus on the number of tests performed. Let $T(i)$ be the expected number of tests performed for updating the existing solution when p_i is obtained. When executing the *RMED_ONLINE* process, a seed performs the test once and, if possible, executes further *RMED* processes. Since the probability that p_i lies in the boundary of D_i is $3/i$, we have the following recurrence.

$$T(i) \leq 1 + \frac{3}{i} T_2(i - 1).$$

Because $T_2(n) \leq 3n$, we achieve $T(i) = O(1)$, that is to say, the updating process *RMED_ONLINE* can be computed in $O(1)$ expected runtime.

4. FALA FOR NON-SEEDS

When seeds have localized their floating areas, we need to utilize them as referees to locate non-seeds. Triangulation from referees is a widely used method to localize static nodes in stationary sensor networks. However, due to the dynamic property, directly using triangulation for RFS leads to poor accuracy. In this section, we propose a new scheme for locating non-seeds based on statistical measurements.

4.1. The Framework of Non-seed FALA

Before looking inside the non-seed FALA, we first define two concepts about computed and computable sensor nodes.

Definition 4.1 (Computed and Computable Sensor Nodes). We call a sensor node computed if its floating area is already known. If a noncomputed sensor node has k ($k \geq 3$) computed neighbors, it is a computable sensor node.

The non-seed FALA is an iterative process, gradually transforming computable sensors to computed sensors. Figure 3 plots a deployment of four sensors: a non-seed s with the floating area o unknown, and its three neighbors $\{s_i \mid 1 \leq i \leq 3\}$. Assume all s_i are computed nodes, that is, their floating area $o_i(c_i, r_i)$ is known. Let d_i denote the distance between s and s_i . Our goal is to estimate the floating area o of s . Clearly, with one time measurement, there exists uncertainty for floating area computation. As shown in Figure 3, another disk area o' different from o is also possible to be a

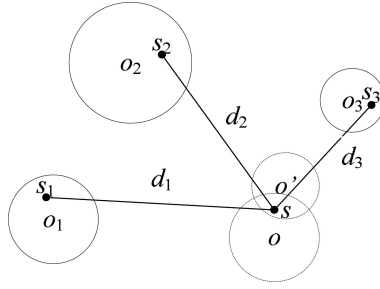


Fig. 3. A non-seed node s and its neighbors.

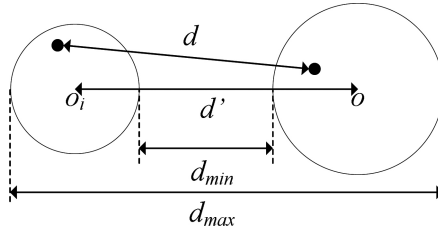


Fig. 4. The distance d between two sensors and d' between two anchored positions.

candidate of the floating area of s , because the current position of s which satisfies all distance constraints also resides in o' . We cannot distinguish the real area from o and o' at this stage. That means, it is impossible to calculate the floating area of s under a single time observation of d_i .

The distance measurement d_i varies all the time due to the movement of s and s_i . We observe that multiple samplings can alleviate the uncertainty for FALA. If we treat o and o' as two sets of points in plane, when s moves to some position in $o - o'$, the distance sampling information negates the possibility of o' being the floating area. Furthermore, we know that d_i only depends on $o(c, r)$ and $o_i(c_i, r_i)$, irrespective of any other floating areas o_j ($j \neq i$). Hence, the sample distribution is determined by r_i , r and the distance between two anchored positions $d'_i = \text{dist}(c, c_i)$, indicating that, to some extent, the sample statistics can imply r_i , r , and d'_i . Therefore, the relationship between the sample statistics and the parameters r_i , r , and d'_i is of great importance, based on which non-seeds can localize their floating areas.

Without loss of generality, we only consider s and a calculated neighbor s_i , as shown in Figure 4. For simplicity, we use d and d' instead of d_i and d'_i to elaborate the non-seed FALA. We know d varies all the time, while d' is a static value. Let D denote the random variable of d , and let d_i denote the observed value of D at sampling time t_i .

Three steps are included in the floating area computation of non-seed s , described as follows.

- (1) A non-seed s samples the distance measurements d between s and its neighbors.
- (2) Based on the sample statistics, s calculates d' and r .
- (3) If s has more than three computed neighbors, it calculates the anchor position c by triangulation based on d' .

In step 1, the non-seed s carries out a sampling process. In step 2, s estimates the hidden parameters based on distance samples. We consider three methods for exploring the relationship between sample statistics and the hidden parameters: the maximum likelihood estimation (MLE), geometrical relationship, and regression analysis.

In step 3, although sensor nodes are mobile, their anchored positions are static. Thus, it is possible to solve a typical point localization problem for locating anchored positions. On the premise that the distances from an unknown anchor position to three known anchored positions are obtained, triangulation can be conducted to calculate the unknown anchored position. With c and r , this step completes the floating area computation of s , and s becomes a computed sensor node.

4.2. Maximum Likelihood Estimation (MLE)

Let $P(D)$ be the probability density function of D . Certainly, d' , r , and r_i are parameters affecting $P(D)$. Among the three parameters, r_i is known because s_i is a computed sensor node. Let $\theta = (d', r)$ denote the unknown parameter vector. Hence, $P(D)$ can be represented as $P(D|\theta)$. According to the observed distribution of sampling data, the non-seed node s aims to predict the unknown parameters. This is actually a parameter estimation problem.

We calculate the likelihood function $L(\theta)$ and let $\hat{\theta}$, which maximizes $L(\theta)$, to be the estimation of unknown θ ,

$$L(\theta) = \prod_{i=1}^n P(d_i|\theta)$$

Since $L(\theta)$ and $\ln(L(\theta))$ reach maximum value simultaneously in parameter space Ω , it is equivalent to maximizing $\ln(L(\theta))$ instead of $L(\theta)$. If $\ln(L(\theta))$ is derivable to θ , $\hat{\theta}$ should satisfy the following.

$$\frac{\partial \ln(L(\theta))}{\partial \theta_i} = 0, i = 1, 2$$

Thus, d' and r can be calculated by solving this equation.

MLE gives us a theoretical method for calculating d' and r . In practice, however, it is difficult to calculate θ from the differential equation, so we cannot rely on MLE in real implementation.

4.3. Geometrical Relationship

A simple method for estimating d' and r is to explore the geometrical relationship between the two floating areas of s and s_i . We define $d_{max} = \max(D)$ and $d_{min} = \min(D)$ as the minimum and maximum values of D , respectively. As shown in Figure 4, d_{max} and d_{min} are obtained in two extreme situations. According to the geometrical relationship, we have

$$d' = \frac{d_{max} + d_{min}}{2},$$

$$r = \frac{d_{max} - d_{min}}{2} - r_i.$$

Such a method is simple to implement and takes little computation cost. Since the extreme cases may not occur in sampling, it is reasonable to regard $\max(d_i)$ and $\min(d_i)$ as the estimation of d_{max} and d_{min} , respectively. Indeed, the sampling quantity and fidelity severely influence the accuracy of estimation. To the best of our knowledge, existing ranging methods, such as RSS- or TDOA-based approaches, usually produce nonnegligible errors which may heavily degrade the effectiveness of the method. On the contrary, the statistical method, based on sampling distribution suffers less from this.

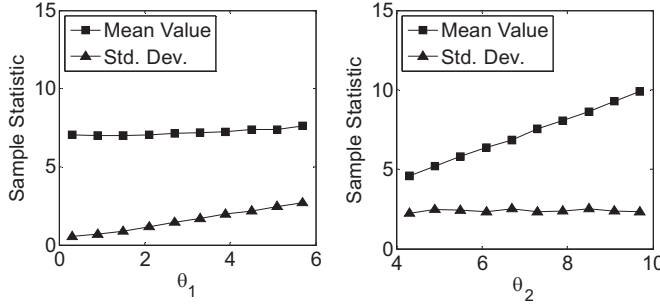


Fig. 5. The relationship between hidden parameters and sample statistics.

4.4. Regression Analysis

As we have observed, the distribution of D , to some extent, reflects the hidden parameters r and d' . This fact allows us to design a method for estimating r and d' based on sample statistics.

In our analysis, since r_i is a known parameter, we introduce two coefficients θ_1 and θ_2 such that $r = \theta_1 \times r_i$ and $d' = \theta_2 \times r_i$. The sample statistics include the mean $\hat{\mu}$ and the standard deviation $\hat{\sigma}$ of samples, defined by

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n d_i,$$

$$\hat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \hat{\mu})^2}$$

A simulation study is conducted to explore the relationship between hidden parameters (θ_1 and θ_2) and sample statistics ($\hat{\mu}$ and $\hat{\sigma}$). We control θ_1 and θ_2 in our simulation. For higher accuracy, $\hat{\mu}$ and $\hat{\sigma}$ are calculated with a large sample capacity.

Figure 5 gives us an intuition about the relationship. When fixing θ_1 , $\hat{\mu}$ grows linearly as θ_2 increases while $\hat{\sigma}$ stays. When fixing θ_2 , $\hat{\sigma}$ grows linearly as θ_1 increase, while $\hat{\mu}$ stays around the value of θ_2 . This observation provides an important hint: there exists a linear relationship between the parameters and the sample statistics.

We now synthetically take account of the impact of both θ_1 and θ_2 by using multiple regression analysis. Let β be a 2×3 coefficient matrix; our general form of a two-variable linear regression equation is as follows.

$$\begin{bmatrix} \hat{\mu} \\ \hat{\sigma} \end{bmatrix} = \beta \times \begin{bmatrix} \theta_1 \\ \theta_2 \\ 1 \end{bmatrix}$$

Using the least squares technique, we have

$$\beta = \begin{bmatrix} 0.0951 & 0.9820 & 0.0409 \\ 0.4507 & 0.0035 & 0.1956 \end{bmatrix}$$

Contrary to our simulation, coefficients θ_1 and θ_2 are unknown in a real deployment. According to the regression model, we calculate θ_1 and θ_2 by the following equation.

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \chi \times \begin{bmatrix} \hat{\mu} \\ \hat{\sigma} \\ 1 \end{bmatrix},$$

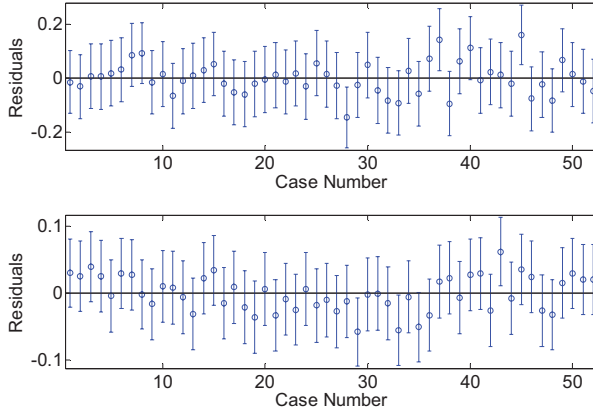


Fig. 6. Residuals of regression analysis.

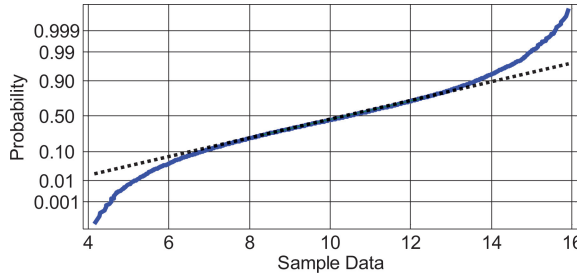


Fig. 7. The normal probability plot of sample data.

where $\chi = \begin{bmatrix} -0.0079 & 2.2204 & -0.4340 \\ 1.0191 & -0.2150 & 0.0004 \end{bmatrix}$.

In summary, a node s first collects sample distances between itself and its neighbor s_i , and then calculates the statistics $\hat{\mu}$ and $\hat{\sigma}$. According to the regression model, s determines θ_1 and θ_2 and finally completes the estimation of r and d' .

Errors of our regression model may come from two sources: (1) the residuals in regression analysis and (2) the inaccuracy of $\hat{\mu}$ and $\hat{\sigma}$. The residual figure, Figure 6, illustrates that the error of our linear regression model is relatively small if we consider the usual values of θ_1 and θ_2 . The inaccuracy of $\hat{\mu}$ and $\hat{\sigma}$ is usually due to a small sample capacity.

Taking error analysis of $\hat{\mu}$ as an example, we want to derive a reasonable sample size with an acceptable accuracy. We first conduct a normality test of sample data. The normal probability plot, Figure 7, suggests that the sampling distribution is normal. Since D is normally distributed with the mean μ , the statistic

$$\frac{\hat{\mu} - \mu}{\hat{\sigma} / \sqrt{n}}$$

possesses a t -distribution with $n - 1$ degrees of freedom, where n is the size of the sample. Then we get

$$P\left(\hat{\mu} - t_{\alpha/2, n-1} \frac{\hat{\sigma}}{\sqrt{n}} < \mu < \hat{\mu} + t_{\alpha/2, n-1} \frac{\hat{\sigma}}{\sqrt{n}}\right) = 1 - \alpha,$$

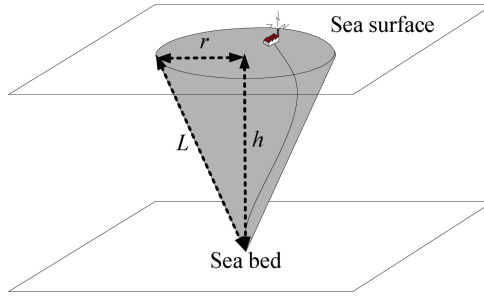


Fig. 8. The geometrical structure of the rope length, radius, and sea depth.

and the interval estimation of μ

$$\left(\hat{\mu} - t_{\alpha/2, n-1} \frac{\hat{\sigma}}{\sqrt{n}}, \hat{\mu} + t_{\alpha/2, n-1} \frac{\hat{\sigma}}{\sqrt{n}} \right)$$

is a $(1 - \alpha) \times 100\%$ confidence interval for the mean μ . The length of the interval is

$$l = 2t_{\alpha/2, n-1} \frac{\hat{\sigma}}{\sqrt{n}}.$$

According to the t -distribution, $n = 30$ deserves a 90% confidence interval with an acceptable accuracy $l = 0.6204 \hat{\sigma}$.

5. SEA DEPTH MEASUREMENT BY FALA

By utilizing FALA, we can efficiently localize the sensor nodes in the network. When we use a rope with length L to anchor the sensor node on the sea of depth h ($L > h$), the sensor node floats within the disk area of radius $r = \sqrt{L^2 - h^2}$, as shown in Figure 8. When the rope becomes straight, the node on the sea surface moves to the boundary of the floating area, while when the rope is not straight, the node simply resides within the floating area. After FALA, we obtain the floating area of a node, achieving its center c as well as its radius r . We can then easily calculate the sea depth at position c . This calculation involves neither extra measurements nor hardware costs. We actually obtain the desired result for free as the byproduct of FALA.

In this work, the main application need is to determine the sea depth of Huanghua Harbor such that we can instantly monitor the extent of siltation on the sea bed of the harbor. While regular sea depth monitoring is desired in high time granularity, the measurement precision is not necessarily high, as long as the measured data allows the port authority to be aware of the siltation extent and to make administrative decisions. Such a harbor is close to the seashore and has relatively shallow water (below ten meter sea depth), so we choose the rope length of L larger than 10 m to guarantee that the sensors float above the sea surface. Later in the experimental evaluation, we test FALA in the sea area off HKUST campus, which is similar to the Huanghua Harbor environment. When the sea depth of a monitoring region is deep, longer ropes are necessary. In this situation, the gravity of ropes cannot be ignored. When sensor nodes are on the boundaries of their floating areas, ropes cannot be straight but form a curve with a steep upper part and a mild lower part. Such a curve can be seen as a part of catenary. We can also calculate the sea depth according to FALA results and the equation of catenary [Beyer 1987].

Figure 9 depicts the system design of FALA. FALA consists of three major steps: data collection, floating area determination, and sea depth estimation. At the first step, seeds and non-seeds collect GPS readings and distance measurements, respectively, for

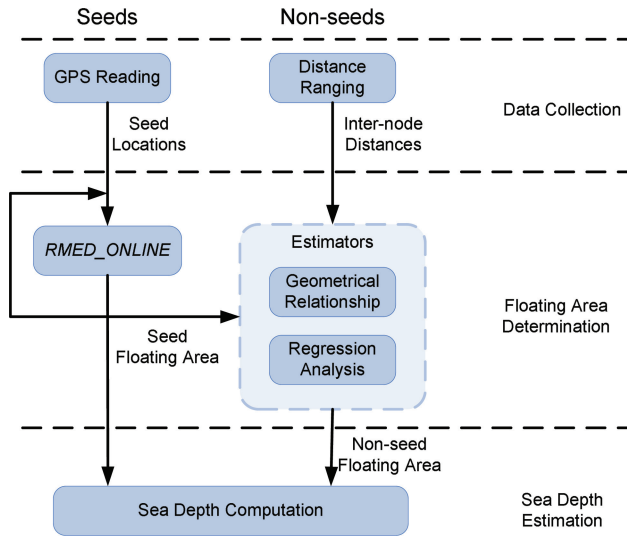


Fig. 9. The system design of FALA.

a certain amount of time until adequate samples are obtained. At the second step, the seeds carry out the RMED_ONLINE procedure to compute their anchored positions and floating areas, while the non-seeds employ two location estimators: geometrical relationship and regression analysis. At the last step, sea depth can be further calculated according to localization results. The accuracy of the final sea depth estimation is largely affected by the precision of the RSSI-based distance measurements.

The entire process of FALA is efficient in terms of energy consumption. When locating non-seeds, distance measurement relies on RSS, which can be obtained through normal data packet delivery, that is, RSS data are free in terms of energy consumption. In addition, current experimental studies have shown that the power used for computation is several orders of magnitude less than that of communication [Polastre et al. 2006]. Thus, neither RMED_ONLINE nor arithmetic operation (for floating area determination of seeds and non-seeds, respectively) requires power resources comparable with communication overhead. Moreover, although sensors float on the sea surface, their floating areas are fixed because they are anchored at fixed positions. Hence, locating floating sensors is a one-time task and consumes relatively less energy compared with other network tasks, such as environment monitoring that requires continuous data collection. To sum up, FALA only contributes a tiny portion of total energy consumption to our system.

6. EXPERIMENTAL EVALUATION

We first examine the effectiveness of our design by deploying a prototype system off the seashore in the HKUST campus. We further conduct a large-scale simulation to test the system scalability under varied network parameters.

We evaluate FALA using three metrics: $E(c) = e_c/r$, $E(r) = e_r/r$, defined in Section 2, and $E(h) = |\hat{h} - h|/h$ to evaluate the error of sea depth measurement. In some previous literature, the location error is represented relative to the hop size (i.e., the maximum communication range of a node) [Goldenberg et al. 2006; Hu and Evans 2004]. However, for FALA evaluation, if we use the communication range as a benchmark to measure location error, a 1m error contributes the same impact to a small floating area as to a large floating area, that is, a 2m radius area and a 10m radius area. To diminish



Fig. 10. Deployment of the prototype system.

this unfairness, we adopt the relative error against the radii of floating areas in the evaluation. Since the communication range of each sensor node is usually 5~15 times larger than the radius of its floating areas in our experiment, the estimate errors shown in this section are usually several times that of what they are against the sensor communication range.

6.1. Prototype Experiment

To better understand the systematic behaviors of FALA, we deploy a prototype with 25 nodes off the seashore on the university campus. The hardware layer of the prototype is constructed on the Telos motes with an Atmel128 processor and CC2420 transceiver. We fit each node with a lightweight supporting shelf, which floats on the sea surface and raises the sensor node 150 cm high above the sea surface. Twenty five such assembled floating nodes are anchored on a 100m \times 100m sea area where the water depth is around 4~7 m. We use 8m rope length accordingly, resulting in a 2~6m floating radius of sensors. Five sensors are selected to be anchor nodes. Figure 10 exhibits our deployment.

We utilize RSSI values from the transceivers to estimate the distances between nodes. The transmitting power of sensor nodes is set to 1 mW, and the transmitting range could reach as far as 40 m with more than -95 dbm receiving signal strength. We construct a distance estimator according to the most widely used signal propagation model: the log-normal shadowing model [Seidel and Rappaport 1992]. Due to the coarse and nonmonotone correspondence between the RSSI and distance in the real measurements, the relative error of the distance estimation could be up to 150%, which heavily limits the accuracy of FALA. We believe more precise distance estimating techniques, such as TDOA - or TOA-based approaches, would help to achieve better FALA accuracy.

Figure 11 plots the FALA performance in our prototype system. The error of the anchored position, as shown in Figure 11(a), is around 0.5~1 for seeds and 0.5~4 for non-seeds. For radius estimation, the error is around 0.05~0.3, illustrated in Figure 11(b). In Figure 11(c), we can see the relative error of sea depth is around 0.03~0.2. From Figure 11, seeds basically outperform non-seeds in all three metrics.

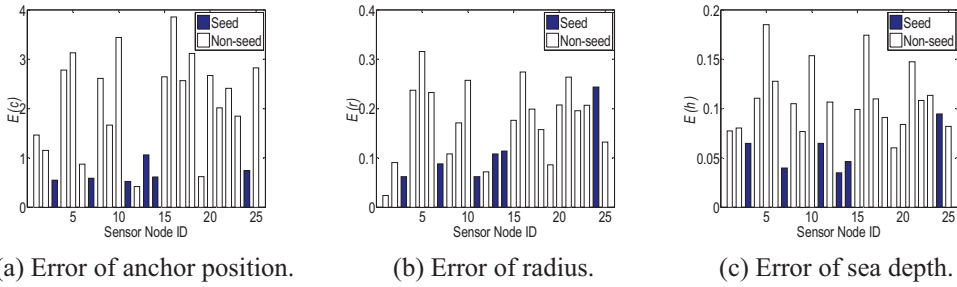


Fig. 11. FALA performance of each node in the prototype system.

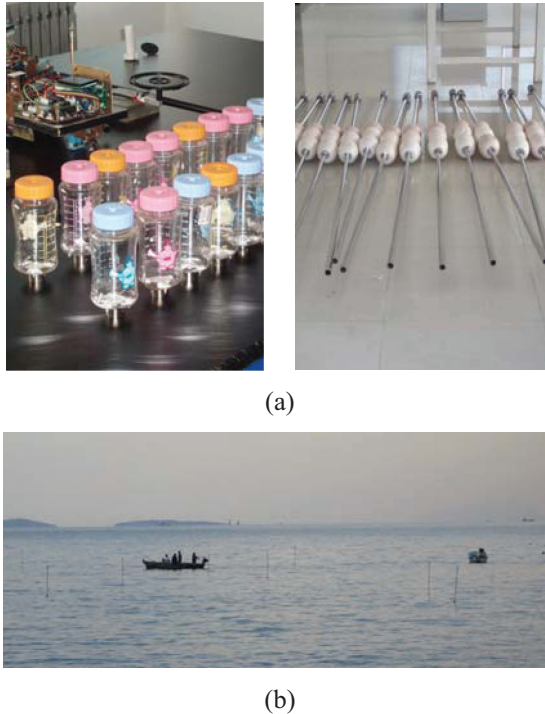


Fig. 12. OceanSense. (a) Assembled sensor nodes; (b) system deployment on the sea.

In practice, two factors limit our prototype from more accurate results: (1) the ocean current near the seashore undergoes more regular ways than affected by random factors, which makes errors on our floating model assumptions; (2) the large errors in our RSSI-based ranging technique contributes much to the estimation error of FALA.

We are currently conducting the OceanSense project, implementing a working system with a larger deployment scale in Tsingtao, China. Figure 12(a) shows our manufactured sensor node with a Telos mote encapsulated in the container and empowered with a high-gain antenna. The plastic encapsulation protects the inside mote from the sea water corrosion which helps to persist the system over a long time. The high-gain antenna helps to provide stable signal strength and increase the RSSI-based ranging accuracy. We attach buoys on the sensor node to let them float on the sea, estimating the sea depth and measuring other environmental elements, such as temperature, light

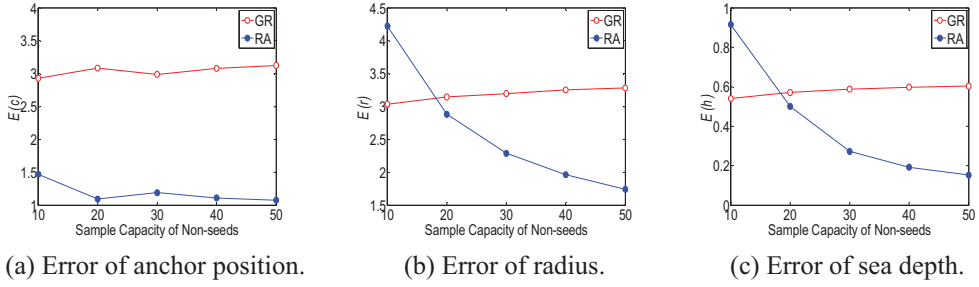


Fig. 13. FALA performance vs. sample capacity with precise distance measurement.

illumination, etc. A Stargate board attached with an IRIS node is deployed among the Telos motes. The Stargate acts as a sink node, receiving information from the sensor nodes and connecting to the Internet through wireless GPRS network. We implement a backend database system storing and processing the information delivered back from Stargate. Figure 12(b) shows the system deployed on the field. We hope that the long-term operational experience from this practical system examines the validity of our approach and that the collected real data trace will provide us with insight knowledge of the real environment variations. A demo video is available on the Web showing how we design the nodes and deploy them on the sea.

6.2. Large-Scale Simulation

In this section, we evaluate the scalability of FALA. We generate networks of 900 nodes randomly distributed in a square sea region. In our simulation, the sea region is designed to be a $600\text{m} \times 600\text{m}$ square and has a water depth around 10 m. When an RFS network is deployed in this sea region, the radii of floating areas are 2~6 m, which are determined by sea depth and rope length. A typical communication range of the sensor nodes is 30 m, and the average degree of network topology is 8. In all our measurements, we integrate the results from 100 instances.

In our simulation, we varied two parameters: the proportion of seeds and the sample capacity to examine FALA under different settings. We test the performances of the geometrical relation method (GR) and the regression analysis method (RA) proposed in Section 4.

Precise Distance Measurements. We first assume precise distance measurement to explore the ideally achievable accuracy of FALA. Figure 13(a) plots the average error of the anchored positions. The error of RA is below 1.5, which is lower than GR as the sample capacity varies in a wide range. When the size of the sample is larger than 20, the extra gain from RA becomes trivial. Therefore, 20 could be a good choice of sample capacity considering the trade-off between accuracy and overhead. As shown in Figure 13(b), the average radius error of RA consistently decreases as the sample size increases, while GR is getting slightly worse. RA outperforms GR when the sample capacity is larger than 20. The average error of sea depth, investigated in Figure 13(c), follows a similar pattern as that of the radius estimation.

We also examine the impact of the seed density on FALA, highlighted in Figure 14. The sample size is set to 20. All performance metrics get better when the seed density increases. There is a notable gap between GR and RA in Figure 14(a). The error of RA is less than 2 when 25% of the seeds exist. In Figure 14(b), we examine the radius estimation. We observe that both RA and GR yield smaller errors when inserting more seeds and that RA is better than GR when seed proportion is larger than 20%. Figure 14(c) shows the error on sea depth measurements. Again, it follows the similar

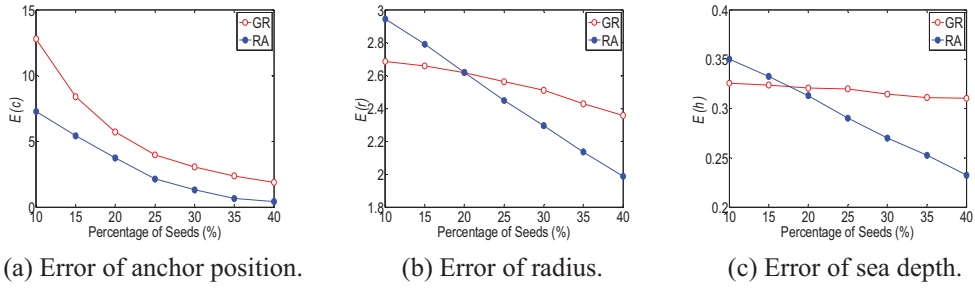


Fig. 14. FALA performance vs. seeds proportion with precise distance measurement.

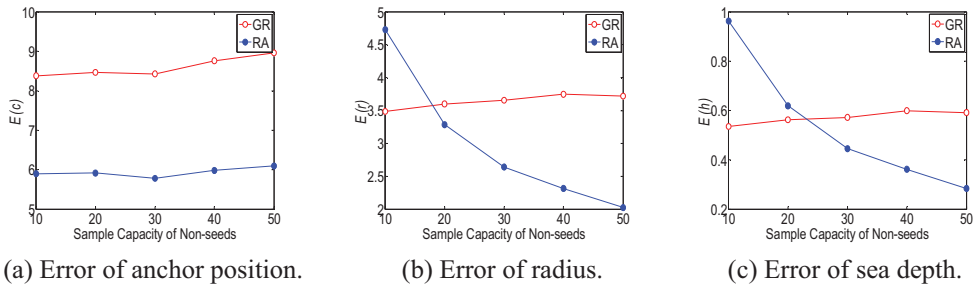


Fig. 15. FALA performance vs. sample capacity with noisy distance measurement.

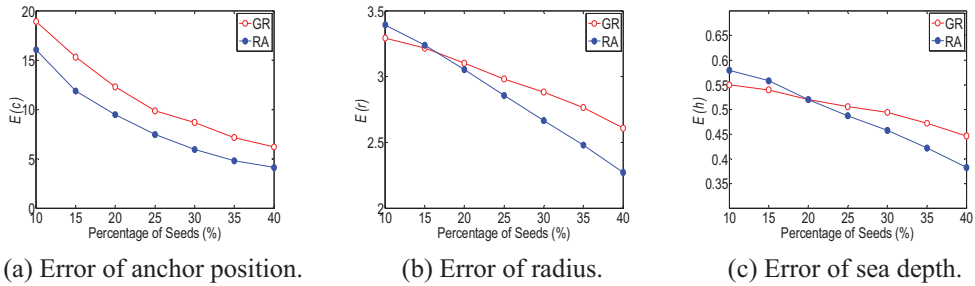


Fig. 16. FALA performance vs. seeds proportion with noisy distance measurement.

pattern as that of radius error, and RA yields the error from 0.35 to 0.2 when the seed proportion varies from 10% to 40%.

Noisy Distance Measurements. We further evaluate FALA under noisy distance measurements. In our simulation, we introduce a zero-mean Gaussian noise with standard deviation of 50% of the real values into distance measurements.

Again, RA outperforms GR, as shown in Figures 15 and 16. Compared with Figures 13 and 14, the error of all performance metrics is larger than the corresponding errors with precise distance measurement. Especially, for anchored position estimation, the error can be three times larger. Clearly, a noisy distance measurement heavily degrades the performance of FALA. In such situations, 30 sample datum and 30% seeds are necessary for precise localization.

7. RELATED WORK

Recent advances in WSNs have attracted the attention of a lot of researchers [Ahn et al. 2006; Karenos and Kalogeraki 2006; Li and Liu 2009; Liu et al. 2011; Ni et al. 2003; Xing et al. 2005], with many efforts towards locating sensors [Goldenberg et al. 2006; He et al. 2003; Li et al. 2011]. A general overview of state-of-the-art location sensing systems is available in Hightower and Borriello [2001]. According to the targeted environments, previous localization approaches can be classified into two types: those for static sensor networks and those for mobile sensor networks.

The static localization problem has been extensively studied for WSNs. The proposed localization approaches typically use a small number of seed nodes that are aware of their location. Moreover, ranging measurements [Bahl and Padmanabhan 2000; Goldenberg et al. 2006; Liu et al. 2006; Niculescu and Nath 2003] (in range-based approaches) or neighborhood information [Bulusu et al. 2000; He et al. 2003; Li and Liu 2010; Niculescu and Nath 2003] (in range-free approaches) are utilized to locate non-seed nodes. All these approaches assume the invariability of sensor locations. Once a sensor node knows its location, it can be used as a beacon to locate other sensor nodes. Such a strategy fails in our RFS context due to the movement of sensors. Some of the static localization approaches [Niculescu and Nath 2003; Savvides et al. 2001] can be extended to conform to the mobile environment. Most of them, however, cannot yield results in real time and thus suffer from estimation latency and inaccuracy brought on by sensor movements. Bergamo and Mazzini's [2002] produced one of the first works related to the localization problem in mobile sensor networks. Two fixed seeds are assumed transmitting across the entire network, and other nodes can measure the received signal strength accurately. Hu and Evans propose a statistic-based localization approach for mobile sensor networks [2004] based on the MCL method [Dellaert et al. 1999], which originates from a mobile localization problem in robotics. Li et al. consider the restricted mobility of sensors along the underground cables, which shares some similarities with this work [2008]. Nevertheless, in their work, the restricted sensors do not move within disk areas, as they do in this work. Mobility creates obstacles to accurate localization, resulting in large errors and heavy communication cost. In addition, dense seed deployment is required in that proposed approach.

8. CONCLUSIONS AND FUTURE WORK

In this work, we discuss a novel sea depth measurement application using wireless sensor networks. We define the localization problem in RFS networks and introduce the concept of floating area localization, so as to determine the floating areas of sensor nodes. A statistical approach, FALA, is designed, based on which the sea depth can be acquired without expensive sonar systems. A prototype with 25 Telos nodes is deployed on a sea surface, and intensive large-scale simulations are conducted to examine the efficiency and scalability of the proposed approach.

This work is still at an initial stage. Future work leads the following directions.

- (1) One assumption in our floating model is that the sensors float within their anchored areas under randomness, which in our prototype test is shown to be inadequate. The seawater near the seashore moves regularly a little rather than being completely affected by randomness. The wave may also introduce errors of estimation. Given such limitations, we believe a good model of the behaviors of the sea would help diminish their negative impact or even make use of their regularity to achieve more accuracy [Guo et al. 2010].
- (2) The system scalability is also an important issue that we need to pay special attention to. Since the RSSI-based distance measurement bears a large error, there is a trend of error propagation on our estimations when the network size significantly

increases, especially under a small percentage of seeds. Whether or not we are able to design a sound-collaborating mechanism at the layer of network topology so that we can suppress the localization errors throughout the network is a significant but challenging issue.

- (3) Sea depth estimation is of great interest and importance for many sea monitoring applications. Our FALA approach yields the estimations of sea depth by utilizing the result of the floating area localizations, thereby reducing the cost. This approach, however, also has its own limitations, for example, the anchor of each sensor can actually get buried by the silt, which leads to inaccurate estimations as time passes, or the RSSI-based localization may introduce errors. Nonetheless, due to the intensive needs on the sea depth measurement and the difficulty of employing infrastructures at sea, we believe WSN is one of the best candidates for this application. Other distance measurement approaches, like TDOA- or TOA-based ones, may help to achieve better accuracy.
- (4) The Restricted Floating Sensors describe a general model for sensor deployment which might be suitable for many sensing applications carried out on the sea. Under different contexts of the sensing applications, we might consider different factors of the network besides the locations, such as sensor coverage, network connectivity, data samplings, etc. Due to the nature of restricted mobility, the RFS network introduces the intermediate dynamics between the static network and mobile network. By developing mechanisms over the dynamics but taking advantage of the restriction on the mobility, could we achieve higher efficiency? We believe it is nontrivial and highly related to the concerned factors and the application context.

REFERENCES

- AHN, G.-S., MILUZZO, E., CAMPBELL, A. T., HONG, S. G., AND CUOMO, F. 2006. Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
- BAHL, P. AND PADMANABHAN, V. N. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*.
- BERGAMO, P. AND MAZZINI, G. 2002. Localization in sensor networks with fading and mobility. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*.
- BEYER, W. H. 1987. *CRC Standard Mathematical Tables*. CRC Press, Boca Raton, FL.
- BULUSU, N., HEIDEMANN, J., AND ESTRIN, D. 2000. GPS-less low cost outdoor localization for very small devices. *IEEE Pers. Commun. Mag.*
- DE BERG, M., KREVELD, M. V., OVERMARS, M., AND SCHWARZKOPF, O. 2000. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin Heidelberg.
- DELLAERT, F., FOX, D., BURGARD, W., AND THRUN, S. 1999. Monte Carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- GOLDENBERG, D., BIHLER, P., CAO, M., FANG, J., ANDERSON, B., MORSE, A. S., AND YANG, Y. R. 2006. Localization in sparse networks using sweeps. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*.
- GUO, D., WU, J., CHEN, H., YUAN, Y., AND LUO, X. 2010. The dynamic bloom filters. *IEEE Trans. Knowl. Data Eng.* 22, 1, 120–133.
- HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. F. 2003. Range-free localization schemes in large scale sensor networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*.
- HIGHTOWER, J. AND BORRIELLO, G. 2001. Location systems for ubiquitous computing. *IEEE Comput.* 34, 8.
- HU, L. AND EVANS, D. 2004. Localization for mobile sensor networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*.
- KARENOS, K. AND KALOGERAKI, V. 2006. Real-time traffic management in sensor networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*.
- LI, M., CHENG, W., LIU, K., HE, Y., LI, X., AND LIAO, X. 2011. Sweep coverage with mobile sensors. *IEEE Trans. Mobile Comput.* 10, 1, 1534–1545.

- LI, M. AND LIU, Y. 2010. Rendered path: Range-free localization in anisotropic sensor networks with holes. *IEEE/ACM Trans. Netw.* 18, 1, 320–332.
- LI, M. AND LIU, Y. 2009. Underground coal mine monitoring with wireless sensor networks. *ACM Trans. Sensor Netw.* 5, 2.
- LI, S., WANG, X., LI, M., AND LIAO, X. 2008. Using cable-based mobile sensors to assist environment surveillance. In *Proceedings of the IEEE International Conference on Parallel and Distributed Systems (ICPADS)*.
- LIU, J., ZHANG, Y., AND ZHAO, F. 2006. Robust distributed node localization with error management. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiHoc)*.
- LIU, Y., ZHU, Y., AND NI, L. M. 2011. A reliability-oriented transmission service in wireless sensor networks. *IEEE Trans. Parallel and Distrib. Sys.* 22, 2100–2107.
- NI, L. M., LIU, Y., LAU, Y. C., AND PATIL, A. 2003. LANDMARC: Indoor location sensing using active RFID. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communication (PerCom)*.
- NICULESCU, D. AND NATH, B. 2003. Ad hoc positioning system (APS) using AOA. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*.
- NICULESCU, D. AND NATH, B. 2003. DV based positioning in ad hoc networks. *J. Telecommu. Syst.*
- POLASTRE, J., SZEWCZYK, R., AND CULLER, D. 2006. Telos: Enabling ultra-low power wireless research. In *Proceedings of the IEEE/ACM Conference on Information Processing in Sensor Networks (IPSN)*.
- SAVVIDES, A., HAN, C., AND SRIVASTAVA, M. B. 2001. Dynamic fine-grained localization in a ad-hoc networks of sensors. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*.
- SEIDEL, S. Y. AND RAPPAPORT, T. S. 1992. 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE Trans. Antennas Propagation* 40, 209–217.
- WELZL, E. 1991. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*. Lecture Notes in Computer Science, vol. 655, Springer-Verlag, Berlin, Heidelberg, 359–370.
- XING, G., LU, C., ZHANG, Y., HUANG, Q., AND PLESS, R. 2005. Minimum power configuration in wireless sensor networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiHoc)*.
- YANG, Z., LI, M., AND LIU, Y. 2007. Sea depth measurement with restricted floating sensors. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*.

Received June 2008; revised August 2010, December 2011; accepted February 2012