A Novel Distributed Sensor Positioning System Using the Dual of Target Tracking

Liqiang Zhang, *Member*, *IEEE*, Qiang Cheng, *Member*, *IEEE*, Yingge Wang, and Sherali Zeadally, *Senior Member*, *IEEE*

Abstract—As one of the fundamental issues in wireless sensor networks (WSNs), the sensor localization problem has recently received extensive attention. In this work, we investigate this problem from a novel perspective by treating it as a functional dual of target tracking. In traditional tracking problems, static location-aware sensors track and predict the position and/or velocity of a moving target. As a dual, we utilize a moving location assistant (LA) (with a global positioning system (GPS) or a predefined moving path) to help location-unaware sensors to accurately discover their positions. We call our proposed system Landscape. In Landscape, an LA (an aircraft, for example) periodically broadcasts its current location (we call it a beacon) while it moves around or through a sensor field. Each sensor collects the location beacons, measures the distance between itself and the LA based on the received signal strength (RSS), and individually calculates their locations via an Unscented Kalman Filter (UKF)-based algorithm. Landscape has several features that are favorable to WSNs, such as high scalability, no intersensor communication overhead, moderate computation cost, robustness to range errors and network connectivity, etc. Extensive simulations demonstrate that Landscape is an efficient sensor positioning scheme for outdoor sensor networks.

Index Terms—Wireless sensor networks, localization algorithm, Unscented Kalman Filter.

1 Introduction

Recent years have witnessed explosive interest in wireless sensor networks (WSNs) both from industry and academia. It is believed that WSNs will revolutionize the way in which we understand and construct complex physical systems [18] and extend our sensory capability to every corner of the world [12]. Future WSNs may consist of hundreds to thousands of sensor nodes communicating over a wireless channel, performing distributed sensing and collaborative data processing tasks for a variety of vital military and civilian applications [1]. In most applications, it is important for the sensor nodes to be aware of their own locations. The usefulness of sensed data without spatial coordinates may be highly reduced. Location-aware sensors may also help to highly enhance the efficiency of routing protocols [30], [31] by reducing costly message flooding. However, installing a global positioning system (GPS) [22] receiver on each sensor node may not be a practical solution for most applications because of the size, the battery, and the cost constraints of sensor nodes.

Localization has been studied for many years as a classical problem in many disciplines, including navigation systems (VOR [54]) and GPS [22]), the robot localization problem in mobile robotics [15], [24], [29], user location identifying in cellular networks [16], [9], [10], and WLANs [2], [43], [53]. However, solutions for the above problems may not be directly applicable to networked sensors. Recently, the sensor location discovery problem has attracted extensive research efforts. It remains a challenging problem, however, because several characteristics are highly desirable for sensor localization schemes and need to be satisfied simultaneously [12]:

1. The positioning algorithm should be distributed in order to scale well for large sensor networks.
2. The localization protocol must minimize the communication and computation overhead for each sensor since nodes have very limited resources (power, CPU, memory, etc.).
3. The positioning functionality should not increase the cost and complexity of the sensor much since an application may require thousands of sensors.
4. A location detection scheme should be robust. It should work with sufficient precision in various environments and should not depend on sensor-to-sensor connectivity/density in the network.

Most research on sensor positioning exploits distance or angle measurements from anchor nodes (with GPS or preset location) or neighbors. When the percentage of anchor nodes (among total nodes) is high enough that each node has three anchor nodes (noncollinear) in its neighborhood, then the localization problem is reduced to simple trigonometric calculations. To minimize the deployment cost, however, researchers are more interested in solutions that

- L. Zhang is with the Department of Computer and Information Sciences, Indiana University South Bend, Northside Hall 331, 1700 Mishawaka Ave., South Bend, IN 46615. E-mail: liqzhang@iusb.edu.
- Q. Cheng is with Siemens Medical Solutions, Siemens Corp., 550 College Road, Princeton, NJ 08540. E-mail: qiangcheng6@yahoo.com.
- Y. Wang is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202. E-mail: am3422@wayne.edu.
- S. Zeadally is with the Department of Computer Science and Information Technology, University of the District of Columbia, 4200 Connecticut Avenue, Washington, DC 20008. E-mail: szeadally@udc.edu.

Manuscript received 7 Nov. 2005; revised 11 July 2006; accepted 23 Feb. 2007; published online 6 Aug. 2007.
Recommended for acceptance by S. Iyengar.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0399-1105.
Digital Object Identifier no. 10.1109/TC.2007.70792.

assume only a small fraction of anchor nodes [3], [6], [7], [14], [25], [47], [48] or even assume that they are anchor-free [20], [44]. Location discovery for these cases has to rely more on the node-to-node distance or angle measurements and the problem itself, in effect, becomes a subset of the geometric graph embedding problem [5], [17], [34], [44], [50] or, more generally, a constrained optimization problem [3], [14], [48]. However, obtaining the optimal solution (estimated locations), in the context of sensor networks, is challenging. Previous proposals have to make trade-offs among accuracy, computation overhead, communication overhead, scalability, and other issues. For example, collecting all constraints (measurements) and resolving the optimization problem centrally [14] may involve high computation complexity, becoming usable only when the application permits the deployment of a central processor to perform location estimation [41]; on the other hand, some distributed solutions (such as AFL [44], which uses a mass-spring model for refinement) may not be able to avoid high-volume interactive communications among neighbors. Furthermore, localization methods based on neighborhood measurement begin to perform acceptably only when the node densities are *well beyond the density required for network connectivity* [13].

In this work, we propose a novel sensor positioning solution named Landscape. It is designed for outdoor sensor networks. In Landscape, by introducing a mobile location assistant (LA, which could be an aircraft, a balloon, a robot, a vehicle, etc.), we investigate the localization problem from a different perspective by taking it as a functional dual of target tracking. Traditional tracking problems utilize one or more static location-aware sensors to track and predict the position (and/or velocity) of a moving target. In Landscape, we let each location-unaware sensor discover its position by passively observing a moving location-aware LA (with a GPS or predefined moving path). We resolve this functional dual problem by utilizing an Unscented Kalman Filter (UKF)-based algorithm [28]. One scenario frequently mentioned in the literature is that sensor nodes are deployed by an aircraft [38]. Landscape fits well with (but is not limited to) this kind of sensor applications. We can simply let the aircraft cruise several rounds above the sensor field, broadcasting beacons periodically while flying. Each beacon contains the aircraft's current location. Sensors collect the beacons, measure the distance between itself and the LA based on the received signal strength (RSS), and individually "track" its own position through the proposed UKF-based algorithm. Landscape has significant benefits compared to other localization methods for outdoor sensor networks:

- *Landscape is a distributed and localized scheme.* Each node discovers its location with its own measurements and calculations. It is applicable to large-scale sensor networks.
- LA broadcasts beacons and sensor nodes listen passively. *There are no interactive intersensor communications involved in this process, which not only saves the sensor's energy, but also lessens the channel congestion.*
- *Landscape uses UKF-based algorithms that introduce only moderate computational cost*, compared to other sequential techniques such as Monte Carlo-based methods [15], [25], [29]. Landscape provides cost-accuracy flexibilities. Each sensor node may use less beacons (less computations) for a lower accuracy or vice versa.
- *Landscape is robust to range errors.*
- *Landscape is independent of network density and topology* and can be used for sensor networks deployed in complex outdoor environments.

Extensive simulations have been conducted to study the performance of Landscape. We have used MDS-MAP [51], a state-of-the-art sensor localization approach, as the reference. Simulation results demonstrate the effectiveness and advantages of Landscape.

The rest of this paper is organized as follows: The next section presents related work. We describe our proposed approach in Sections 3 and 4. The performance evaluations of our proposed system are presented and discussed in Section 5. We discuss some possible extensions to further enhance the performance of Landscape in Section 6. Finally, Section 7 presents our concluding remarks.

2 Related Work

Sensor localization has attracted significant research effort in recent years. Many approaches have been proposed. The majority of them assume that a small fraction of the nodes (called anchors, beacons, or landmarks) have a priori knowledge of their locations. Most of them also follow a common process for location discovery. The first phase is to estimate the distances or angles to anchors or other neighboring nodes, which is often called ranging. The second phase is to estimate the positions based on the ranging measurements. Some proposals have an optional third phase, which is to refine the position estimation utilizing the local [48], [51] or global information [51]. There are different ways to categorize the existing approaches by the techniques used in these phases. In this section, we give a brief literature review. First, we classify the existing methods into three groups according to the ranging techniques (time of arrival/time difference of arrival (ToA/TDoA), angle of arrival (AoA), and RSS); we use a separate section to describe MDS-MAP-related algorithms [51. Then, we take a glance at Bayesian techniques for robot location estimations. Finally, we briefly discuss some approaches that utilize mobile beacons.

2.1 Location Discovery with T(D)oA Measurement

RF ToA is a common technology used to measure distance via signal propagation time. Since an RF signal travels 0.3 m/ns, using ToA requires precise clock synchronization between the transmitter and receiver when one-way measurement is used and, even when two-way measurement (round-trip time) is taken, results may still be highly biased by the timing inaccuracy caused by any processing delays [41]. Multipath and nonline of sight (NLOS) are two other sources of errors for ToA measurements. The most basic localization system using ToA is the GPS [22], which requires relatively expensive and energy-consuming electronics to precisely synchronize with a satellite's clock [20].

TDoA uses the difference between the traveling times of two signals to estimate distance. For example, in AHLoS [47], Savvides et al. measure the time difference between two simultaneously transmitted radio and ultrasound at the receiver. Generally, TDoA gives more accurate distance estimations than ToA since the second medium (for example, ultrasound) travels at a much slower speed, making it not as sensitive to timing as ToA. However, equipping ultrasound to a sensor node not only means more cost and energy consumption but also requires nodes to be densely deployed (ultrasound usually only propagates to 20-30 feet), which makes it unsuitable for outdoor applications.

Čapkun et al. [11] proposed a "GPS-free" positioning scheme for mobile ad hoc networks which utilizes the ToA method to measure the distances between each node and its neighbors. AHLoS [47] is a well-known TDoA-based sensor localization system in which nodes estimate their locations by iterative multilateration and collaborative multilateration. Cheng et al. proposed TPS in [12], a novel time-based positioning scheme for outdoor sensor networks which utilizes the TDoA technique in a different way than the one used by Savvides et al. In TPS, TDoA is used to measure the time difference between two radios relayed through different paths, instead of the time difference between one radio and one ultrasound traveling through the same path.

2.2 Location Discovery with AoA Measurement

In AoA measurement, directive antennas or antenna arrays are used to estimate the angle of arrival of the received signal from a beacon node. The concept of AoA was originally used in the VOR/VORTAC system [54] for aircraft navigation. When used for sensor positioning, two factors have to be considered: 1) AoA can be difficult to measure accurately if a sensor is surrounded by scattering objects [10] and 2) the required directive antennas or antenna arrays for AoA measurement may become prohibitive for tiny or cheap sensor nodes.

Niculescu and Nath proposed an AoA-based ad hoc positioning system [38]. To avoid using directive antennas or antenna arrays, each sensor node is equipped with two ultrasound receivers placed at a known distance from each other to calculate the angle of arrival, a technique developed by the Cricket Compass project [45] at the Massachusetts Institute of Technology. A directionality-based scheme was proposed by Nasipuri and Li in [36] in which they transferred the need for directive antennas from the sensors to the anchors by a delicate design.

2.3 Location Discovery with RSS Measurement

RSS has been widely used as a distance measure in the context of WSNs because of its simplicity. In the RSS method, the measured received power and the known transmitted power are used to determine the channel path loss. Although the path loss is also affected by unpredictable shadowing and frequency-selective fading, the path loss is highly correlated with the path length [40]. RSS-based localization has been studied extensively [2], [3], [26], [37], [40], [41], [43], [55]. An extreme case of the RSS method is connectivity (or proximity) measurement, in which only the information on whether or not two devices are connected or "in-range" is used. The schemes based on connectivity [6], [14], [20], [25], [35], [51] are often referred to as *range-free* algorithms, which provide coarse-grained localization only but show robustness in situations where RSS is hard to estimate accurately, such as in complex indoor or urban environments.

The ad hoc positioning system (APS) [37] first estimates ranges based on DV-hop, DV-distance, or euclidean and then applies trilateration to compute the location of each sensor. Bulusu et al. [6] proposed a range-free algorithm. In this work, anchors broadcast their positions to neighbors that keep an account of all received beacons. Using this proximity information, a simple centroid model is applied to estimate the listening node's location. He et al. [20] proposed another range-free positioning scheme, called APIT, in which the possible a target node resides is narrowed down by a point-in-triangulation (PIT) test.

Patwari et al. [40], [41] studied the Cramer-Rao Bound (CRB) for location estimation based on RSS measurement. Whitehouse and Culler [55], [56] designed and evaluated an ad hoc localization system called Calamari. They had an important observation about RSS measurement: Although it is well known that RSS is unreliable in complex indoor or urban environments, many sensor network applications are situated in ideal settings for measuring RSS, for example, outdoors [55]. Furthermore, they showed in [56] that calibration can be used to highly enhance the accuracy of RSS measurements.

2.4 Multidimensional Scaling (MDS) for Localization

MDS [4] has recently been successfully used to resolve the sensor localization problem [51], [26]. Originating from psychometrics and psychophysics and often used as part of exploratory data analysis or information visualization, MDS can be seen as a set of data analysis techniques that display the structure of distance-like data as a geometrical picture [4] which nicely matches the sensor positioning problem. Shang et al. proposed MDS-MAP to use MDS in solving the sensor location problem in [51]. MDS-MAP is a centralized algorithm which consists of three steps: 1) Compute the shortest paths between all pairs of nodes in the sensor field. The shortest path distances are used to construct the distance matrix for MDS. 2) Apply classical MDS to the distance matrix, retaining the first two (or three) largest eigenvalues and eigenvectors to construct a 2D (or 3D) relative map. 3) Given sufficient anchors (three or more for 2D and four or more for 3D), transform the relative map to an absolute map based on the absolute positions of anchors.

MDS-MAP(P) [51] is a distributed version of MDS-MAP. In MDS-MAP(P), individual nodes compute their own local maps using their local information (the range of the local map may contain one-hop or two-hop neighbors) and, then, the local maps are merged to form a global map. If an optional refinement process is used for the global map, the algorithm is called MDS-MAP(P,R).

2.5 Bayesian Techniques for Robot Localization

Bayesian techniques have been widely investigated in the context of robot localization [19], [24]. Recently, grid-based Markov localization [8], particle filtering (PF, also known as sequential Monte Carlo (SMC)) [15], and real-time particle filters [29] have been proposed and shown to be successful for robot location estimation. These Bayesian techniques generally require intensive computation power. There are substantial differences between robot localization and sensor node positioning. First, while robot localization locates a robot in a predefined map, localization in sensor networks works in a free space or unmapped terrain. Second, while a robot can acquire accurate ranging and orientation measurements to landmarks simultaneously with relatively expensive equipment, small sensor nodes cannot. Third, a robot has much more computation power than a sensor node and is able to execute complicated location algorithms.

While our work was in progress, we noticed an interesting work done by Hu and Evans called Monte Carlo Localization (MCL) [25]. Inspired by the techniques used for robot localization, they first proposed to use the SMC method for mobile sensor node localization. Our work is different from theirs in several aspects: 1) MCL requires a certain percentage of mobile anchors to work well and it is designed for mobile sensor nodes. Landscape needs only one mobile LA and is currently for static sensor networks. 2) MCL utilizes only proximity measurements, yielding only coarse-grained location estimations. In contrast, Landscape exploits range measurements and is able to achieve high accuracy. 3) SMC requires intensive computation power; therefore, upgrading MCL to range measurements might be impractical because that would further highly increase the computation cost.

2.6 Utilizing Mobile Beacons for Localization

A preliminary version of this work has been published in [58]. Recently, we have noticed that some other recent work [32], [39], [46], [52], in parallel to our work, also exploit mobile beacons for sensor localization.

Priyantha et al. proposed an approach called mobile-assisted localization (MAL) [46], in which a mobile user (roving human or robot) wanders through a sensor-monitored area, collecting distance information between the sensor nodes and itself; each node then utilizes the distance constraints to "find" its own location. Differently from our work, the work in [46] focuses on the "global rigidity" problem, that is, how to acquire enough distance measurements so that a "globally rigid" structure could be built (thus, the location of a sensor node could be uniquely identified); the calculation process (of utilizing the measurements to find the sensor location) is not their focus. Sichitiu and Ramadurai [52], Kushwaha et al. [32], and Pathirana et al. [39] all share a similar idea of using a location-aware mobile node to localize location-unaware static nodes but exploit different algorithms for location calculations. To minimize the effect of ranging errors, in all of the approaches mentioned above, including the work of Priyantha et al. [46], a sensor node utilizes only the beacons in its close vicinity.[1] To achieve this goal, a complex moving strategy has to be planned so that all of the sensor nodes are "covered" by enough beacons in their close vicinity. Designing such a moving strategy is a challenging task [46]; a complex moving trajectory through a sensor field may also take a long time to finish and thus may introduce a long delay in the location-finding procedure [39]. In Landscape, however, thanks to the UKF-based algorithm, many more beacons (tens to hundreds) could be utilized to reduce the effect of ranging errors; the LA does not have to visit the close vicinity of each sensor node. A simple moving trajectory (as will be shown in later sections) will work well enough for most applications.

1. Ranging errors are usually proportional to the distance.

3 Landscape Localization Methodology and Model

3.1 Landscape Methodology

In this paper, we design our sensor localization system Landscape with an LA, for example, an airplane, a mobile robot, a vehicle, a balloon, etc. The LA can be the carrier disseminating the sensor nodes. Our key idea is to treat the sensor localization as a functional dual to the target tracking problem. In target tracking, one (or more) location-aware sensor node actively estimates the position (and, optionally, velocity and acceleration) of a moving target based on the measurable distances or AOAs, while the moving target plays a passive role. As a functional dual, in Landscape, each location-unaware sensor node utilizes the measured RSS to estimate its own position, while the location-aware LA actively propagates beacons. From this novel perspective, our Landscape system exploits the varying positions of the LA and the corresponding sensor-to-LA distances to dynamically determine the positions of sensor nodes.

Specifically, we determine the sensor localization based on the RSS. An LA is equipped with a GPS or follows a predefined path so that its instant positions are available. It broadcasts messages via RF to the sensors. Each sensor is equipped with a receiving antenna, which can measure the RSS to dictate its distance to the LA. Then, the sensor position is determined by solving the associated state evolvement and observation dynamics of the positions of the LA and the measured distances. In the following text, we use bold letters to denote vectors or vector functions. The superscript T represents the transpose. $E[\cdot]$ and $Var[\cdot]$ denote the expectation and variance, respectively.

3.2 Landscape Localization Model

We define the state variable as the (unknown) 3D position of a specific sensor node. The system state of the ith ($i = 1, \cdots, I$) sensor node at the nth ($n = 1, \cdots, N$) iteration is $\mathbf{x}_i(n) = [x_{i1}(n), x_{i2}(n), x_{i3}(n)]^T$. Moreover, we have the following dynamic state and observation equations:

$$\begin{aligned} \mathbf{x}_i(n) &= \mathbf{f}(\mathbf{x}_i(n-1)) + \mathbf{w}_i(n), \\ \mathbf{y}_i(n) &= \mathbf{g}(\mathbf{x}_i(n)) + \mathbf{v}_i(n), \end{aligned} \quad (1)$$

where $\mathbf{y}_i(n)$ is the observation vector at the nth iteration for node i; $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$, respectively, are the state evolvement and observation functions. $\mathbf{f}(\cdot)$ may be linear or nonlinear depending on application scenarios, whereas $\mathbf{g}(\cdot)$ is usually highly nonlinear. $\{\mathbf{w}_i(n)\}$ and $\{\mathbf{v}_i(n)\}$ are the state and observation noise sequences. We assume that $\{\mathbf{w}_i(n)\}$ and $\{\mathbf{v}_i(n)\}$ are zero mean and uncorrelated.

Here, let us consider the localization of static sensors, where the positions of sensors remain unchanged after deployment. That is, the state dynamics $\mathbf{f}(\cdot)$ governing the sensor positions are simply the identity functions:

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \mathbf{w}_i(n), \quad (2)$$

with $\{\mathbf{w}_i(n)\}$ modeling the small position perturbation due to the wind or other environmental effects. Our algorithm may be extended to mobile sensors by incorporating time-varying state dynamics, which is one of our future research lines.

The state dynamics on the LA are controlled or programmed in advance. Equipped with an accurate GPS, the LA knows its current location, which may be transmitted through an RF signal to the sensors. The observation vector will be the distance from the LA to a sensor node (measured by the node). The following observation model is used:

$$\mathbf{y}_i(n) = \sqrt{(\Delta x_{i1}(n))^2 + (\Delta x_{i2}(n))^2 + (\Delta x_{i3}(n))^2} + v_i(n). \quad (3)$$

Here, $\Delta x_{i1}(n) = x_1^b(n) - x_{i1}(n)$, $\Delta x_{i2}(n) = x_2^b(n) - x_{i2}(n)$, $\Delta x_{i3}(n) = x_3^b(n) - x_{i3}(n)$, and $\mathbf{x}^b(n) = [x_1^b(n), x_2^b(n), x_3^b(n)]^T$ is the current 3D position of the LA, measured by using GPS or controlled with a predefined path. $v_i(n)$ models the observation error due to the RF distance estimation inaccuracy or LA position perturbations.

4 Landscape State Estimation via Unscented Kalman Filtering

Landscape aims at improving the sensor localization accuracy by iteratively incorporating new information and updating the position estimates with the current observations. For the system model defined in the previous section, an online state estimation needs to be performed. Kalman filters (KFs) and their variants have been designed for this purpose, but their actual performance heavily depends on the evolvement and observation equations, as well as the nature of the noise sequences. Due to the nonlinearity of the observation equation, which is the root sum of squares of position differences, the standard KF is not suitable to our applications. Neither is the extended KF (EKF), the first-order approximation to the nonlinear system that is often plagued by the empirical linearization. For nonlinear observation function $\mathbf{g}(\cdot)$, the unscented transformation (UT) [33], [27] can provide higher order approximations without calculating any derivatives. It accurately captures the statistical mean and variance up to the third order of the Taylor expansion of $\mathbf{g}(\cdot)$ for Gaussian noise or second order for arbitrary noise. Even higher order approximations may be obtained with extended algorithms [28]. These gains come with the same order of calculations as the linearization. We develop our Landscape system using UKF [27] with UT.

4.1 Unscented Transformation

UT has been developed to handle low-order statistics of random vectors that undergo any nonlinear transform $\mathbf{u} = \mathbf{g}(\mathbf{s})$. The knowledge of higher order information can also be partially incorporated into the *sigma point* set. Generally, it is based on an observation that it is easier to approximate a Gaussian distribution than an arbitrary nonlinear function. Let D_s be the dimension of the vector $\mathbf{s}$, μ_s be the mean, and K_s be the variance matrix; the UT calculates the first two moments in the following way:

1. Generate a set of sigma points

   $$\mathcal{S} = \{\mathbf{s}_k, W_k : k = 0, \cdots, 2D_s\},$$

   where $\mathbf{s}_k$ are D_s-dimensional vectors and W_k are weights associated with each $\mathbf{s}_k$. Especially, W_0 denotes the weight on the mean point:

   $$\begin{aligned} \mathbf{s}_0 &= \mu_s, \quad W_0 = W_0, \\ \mathbf{s}_k &= \mu_s + \left(\sqrt{\frac{D_s}{1-W_0} Ks}\right)_k, \\ W_k &= \frac{1-W_0}{2D_s}, \\ \mathbf{s}_{k+D_s} &= \mu_s - \left(\sqrt{\frac{D_s}{1-W_0} Ks}\right)_k, \\ W_{k+D_s} &= \frac{1-W_0}{2D_s}, \end{aligned} \quad (4)$$

   where $(K)_k$ denotes the kth row or column of matrix K, for $k = 1, \cdots, D_s$. It can be observed that $\sum_{k=0}^{2D_s} \mathbf{s}_k W_k = \mu_s$.
2. Propagate the sigma points through the nonlinear transformation $\mathbf{u}_k = \mathbf{g}(\mathbf{s}_k)$, $k = 0, \cdots, 2D_s$.
3. Calculate the mean and variance, μ_u and K_u, of the transformed points:

   $$\mu_u = \sum_{k=0}^{2D_s} W_k \mathbf{u}_k, \quad (5)$$

   $$K_u = \sum_{k=0}^{2D_s} W_k (\mathbf{u}_k - \mu_u)(\mathbf{u}_k - \mu_u)^T. \quad (6)$$

By using this set of sigma points, UT provides a Gaussian approximation to the distributions of predictive state and observation vectors. This procedure can be used recursively in a KF structure.

4.2 Unscented Kalman Filter

UKF embeds UT into the KF's recursive prediction and update structure. We consider a single sensor node with the state vector $\mathbf{x}(k)$, noise vector $\mathbf{w}(k)$, observation noise $\mathbf{v}(k)$, and observation $\mathbf{y}(k)$, $k \geq 0$. Notice that we have dropped the subscript so that it can represent any sensor node to be localized. We expand the state vector with the state and observation noise $\mathbf{x}^a(n) = [\mathbf{x}^T(k), \mathbf{w}^T(k+1), \mathbf{v}^T(k+1)]^T$ and the observation vector with $\mathbf{y}^a(k) = [\mathbf{y}^T(k), \mathbf{v}^T(k)]^T$, $k \geq 0$. The resulting augmented state vector is of dimension $D_a = D_x + D_w + D_v$, whereas the augmented observation is of dimension $D_y + D_v$. The state and observation models for the augmented vector $\mathbf{x}^a(n)$ at iteration n from (1) are

$$\begin{aligned} \mathbf{x}^a(n) &= \mathbf{f}^a(\mathbf{x}^a(n-1)), \\ \mathbf{y}^a(n) &= \mathbf{g}^a(\mathbf{x}^a(n)), \\ & n \geq 1. \end{aligned} \quad (7)$$

UKF is implemented as follows [27]: First, we initialize with $\mathbf{x}^a(0) = [\mathbf{x}^T(0), \mathbf{0}, \mathbf{0}]^T$, $K^a(0) = Blk_{Diag}(K_0, Q, R)$, where Q and R respectively are the variances for $\mathbf{w}(n)$ and $\mathbf{v}(n)$, $Blk_{Diag}(K_0, Q, R)$ represents a block diagonal matrix with K_0, Q, and R on the diagonal, $\mathbf{x}^a(0)$ is the initial state value, and $K^a(0)$ is the initial variance of the augmented state.

Then, we utilize the iterative structure of KF. The nonlinear transformation is handled with UT. For the nth iteration, $n \geq 1$, we have the mean and variance of the augmented state vector at the previous iteration, $\mu_x^a(n-1)$ and $K_x^a(n-1)$, as the input.

First, apply the sigma point procedure to the augmented system (7). Construct an approximation of the state estimation to get the sigma points $\{\mathbf{x}_k^a(n), W_k : k = 0, \cdots, 2D_a\}$. Notice that the subscript use indexes the sigma points for the augmented state variables. This should not be confused with the subscripts in (1) for indexing sensors.

Next, compute the predictive means and variances for the state and observation vectors at time n, based on the previous observations. This can be achieved by using the KF prediction structure and UT is used to deal with the nonlinearity and arbitrary distributions:

$$\mathbf{x}_k^a(n) = \mathbf{f}^a(\mathbf{x}_k^a(n-1)), \quad \mathbf{y}_k^a(n) = \mathbf{g}^a(\mathbf{x}_k^a(n)), \tag{8}$$

$$\mu_x^a(n|n-1) = \sum_{k=0}^{2D_a} W_k \mathbf{x}_k^a(n), \tag{9}$$

$$K_x^a(n|n-1) = \sum_{k=0}^{2D_a} (W_k(\mathbf{x}_k^a(n) - \mu_x^a(n|n-1)) (\mathbf{x}_k^a(n) - \mu_x^a(n|n-1))^T), \tag{10}$$

$$\mu_y^a(n|n-1) = \sum_{k=0}^{2D_a} W_k \mathbf{y}_k^a(n), \tag{11}$$

$$K_y^a(n|n-1) = \sum_{k=0}^{2D_a} (W_k(\mathbf{y}_k^a(n) - \mu_y^a(n|n-1)) (\mathbf{y}_k^a(n) - \mu_y^a(n|n-1))^T), \tag{12}$$

$$K_{xy}^a(n|n-1) = \sum_{k=0}^{2D_a} (W_k(\mathbf{x}_k^a(n) - \mu_x^a(n|n-1)) (\mathbf{y}_k^a(n) - \mu_y^a(n|n-1))^T). \tag{13}$$

In the above equations, $\mu_x^a(n|n-1)$ and $K_x^a(n|n-1)$ are the predictive mean and variance, respectively, for the augmented state at time n, given the past observations up to time $n-1$. Similarly, the predictive mean and variance for the augmented observation are $\mu_y^a(n|n-1)$ and $K_y^a(n|n-1)$, respectively. Moreover, $K_{xy}^a(n|n-1)$ is the predictive covariance between the augmented state and the observation.

The update step uses the predictive means and variances with the new observation $\mathbf{y}^a(n)$ to compute the new state mean and variance:

$$W(n|n-1) = K_{xy}^a(n|n-1)(K_y^a(n|n-1))^{-1}, \tag{14}$$

$$\mu_x^a(n) = \mu_x(n-1) + W(n|n-1)(\mathbf{y}^a(n) - \mu_y^a(n|n-1)), \tag{15}$$

$$K_x^a(n) = K_x^a(n|n-1) - W(n|n-1)K_y^a(n|n-1)W^T(n|n-1). \tag{16}$$

In the above equations, $W(n|n-1)$ is a gain matrix. Now, we have obtained the new mean and variance for the augmented state, which may be used as the input to the next iteration at time $n+1$.

With the same order of calculations as EKF, UKF can approximate the second-order Taylor series expansion for arbitrary distributions. In contrast, EKF can only approximate the first order; therefore, UKF is much more accurate, as demonstrated by many applications [27], [33]. Compared to PF, UKF uses a deterministic set of sigma points instead of a large number of particles. Thus, UKF can be implemented in a well-controlled manner. Because of its implementation simplicity and high accuracy, we exploit UKF for state estimation in our Landscape system.

4.3 Unscented-Kalman-Filter-Based Localization Algorithm

Fig. 1 outlines the UKF-based localization algorithm, which is executed by all sensor nodes. As shown in the picture, the algorithm has an optional calibration procedure, which could be done before sensor nodes are deployed. The purpose of the calibration is to improve the accuracy of RSS-based distance measurements [56]. The core of the algorithm is the iteration of state prediction and updating, which could be done either online or offline. When the iterations are done offline, each sensor node first collects all of the observation pairs (each of which contains a beacon's position and its distance from the sensor node) and then executes the UKF loops to update its location estimation, exploiting the constraints increasingly added by each observation pair. The offline version of the algorithm does not have a time constraint[2] on each iteration of state prediction/updating; thus, it is more suitable for sensor nodes that have lower computation power. However, each sensor node needs to have several kilobytes of memory[3] to temporarily store the observation pairs, which is a reasonable requirement for most sensor applications.

Since each sensor individually calculates its own location, the computation complexity of Landscape is independent of the network size. In another words, the computation overhead is $\bigcirc(n)$ (n is the number of sensors in the network) in terms of the whole network or $\bigcirc(1)$ in terms of each sensor. Unlike neighborhood-measurement-based location methods, where sensors usually communicate with each other massively (for ranging measurements and for exchanging location estimations to refine the results), Landscape introduces zero intersensor communications. Communication from sensor nodes to the LA is not needed

2. The online version requires the iteration for one beacon be finished before the next beacon comes.

3. As shown in a later section, for the example scenarios, 240 beacons are sufficient for Landscape to work well. If we use 4 bytes to represent a beacon's (2D) location and 2 bytes to represent the distance from the beacon to a sensor node, the observation pair for each beacon will consume 6 memory bytes to store. For 240 beacons, we need a total of 1,440 bytes.

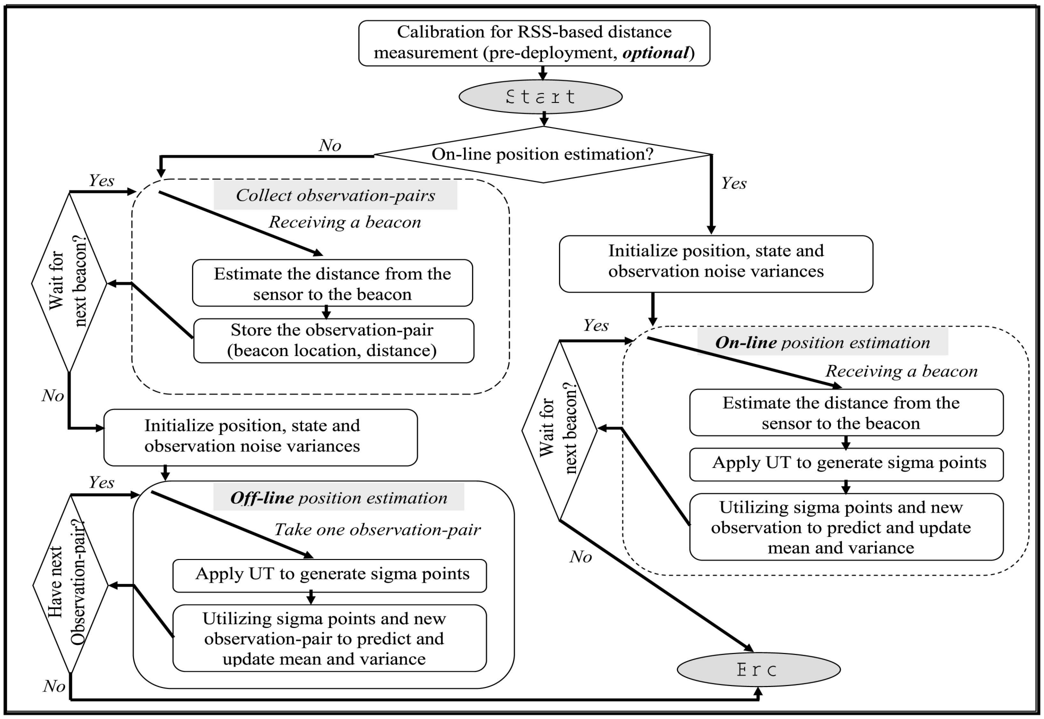


Fig. 1. The UKF-based localization algorithm.

as well. Considering communications usually are more energy-consuming than computations, this is a very attractive feature of Landscape.

4.4 Geometric Dilution of Precision (GDOP) and LA's Moving Strategy

Ranging error is one of the major factors that make the sensor localization a challenging problem. Indeed, an important advantage of Landscape is that multiple distance measurements are effectively and elegantly utilized to compensate for the effect of ranging error on the final position estimation. Besides the ranging error, the geometry of the structure induced by the beacons and each sensor node may also influence the error of the position estimation. The contribution to the estimation error due to geometry is called GDOP [49]. In particular, for Landscape, the effect of GDOP on the final position estimation has a large range. Generally, it decides how fast the iterative estimations converge to the true position. In the worst case, however, it may lead to a totally different position (than the true one). Fortunately, as we will see in the following paragraphs, by choosing a simple moving strategy for the LA, not only can such a worst-case situation be safely avoided, but also a fast convergency to the true positions could be easily acquired.

The effect of GDOP is decided by two factors: the moving trajectory of the LA and the initial value of the position estimation (we call it initial position in the rest of this paper).[4] Fig. 2 illustrates five combinations of these two factors in which cases A and B use a straight-line moving trajectory, cases C and D use a circle moving trajectory, and case E could be considered as a generalized format of case C. Please notice that, instead of enumerating all kinds of curved trajectories, we have used circle-shaped (cases C and D) and arc-shaped (case E) trajectories as representatives. A complex moving trajectory could be considered as a concatenation of multiple arcs.

Obviously, for the straight-line moving trajectory used in cases A and B, the geometry constructed by the mobile beacons and a sensor node does not decide a unique position for the sensor node—the node could be on either side of the line while having the same distance measurements. The estimated position is thus decided by the initial position. Case A represents a situation in which the initial position and the true position of a sensor node are on the same side of the moving trajectory. For this situation, the estimation will quickly converge to the true position. Case B presents the contrary situation, in which the initial position and the true position fall on different sides of the moving trajectory. For this situation, the estimation will finally converge to the **mirror** point of the true position, which is actually the worst case for Landscape.

Differently from the straight-line moving trajectory, a curved LA trajectory will always help sensor nodes to find their unique locations. However, different initial positions may lead to different converging speeds. If the initial position happens to be on the same side as the true position (case C), a sensor node will experience fast convergency (using fewer beacons and less computation power), whereas, if the initial position falls on the other side of the moving trajectory (case D), it will still be able to converge to the true position but will take a relatively longer process. The convergency speed of cases C and D is compared in Fig. 3a through an experiment which is configured as the following: In a $1{,}000 \times 1{,}000$ 2D sensor field, the LA's moving trajectory is a circle with center $(500, 500)$ and radius 500; without losing generality, $(300, 600)$

4. For simplicity, all sensor nodes could share the same initial value, which could be broadcast to them by the LA.

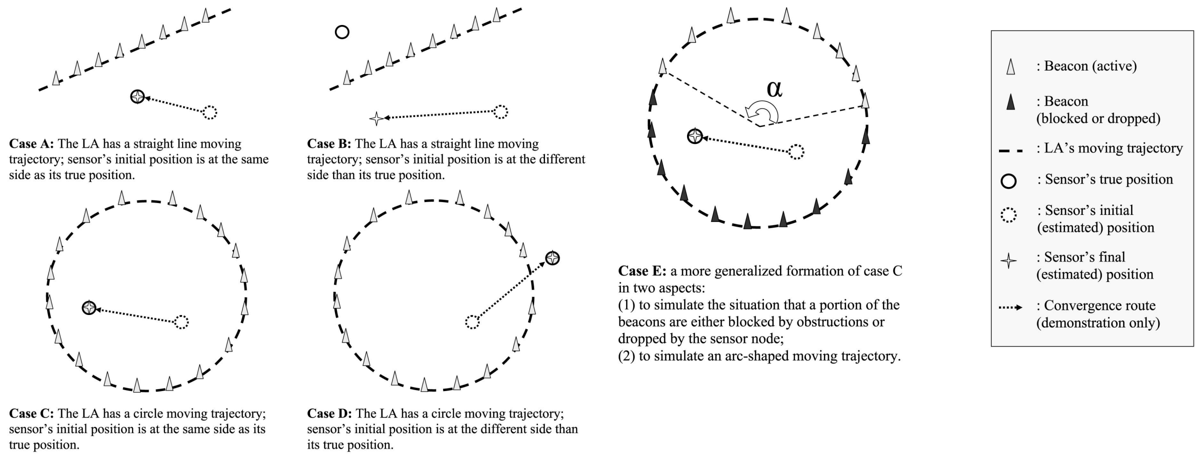


Fig. 2. GDOP versus LA's moving strategy and the sensor's initial position.

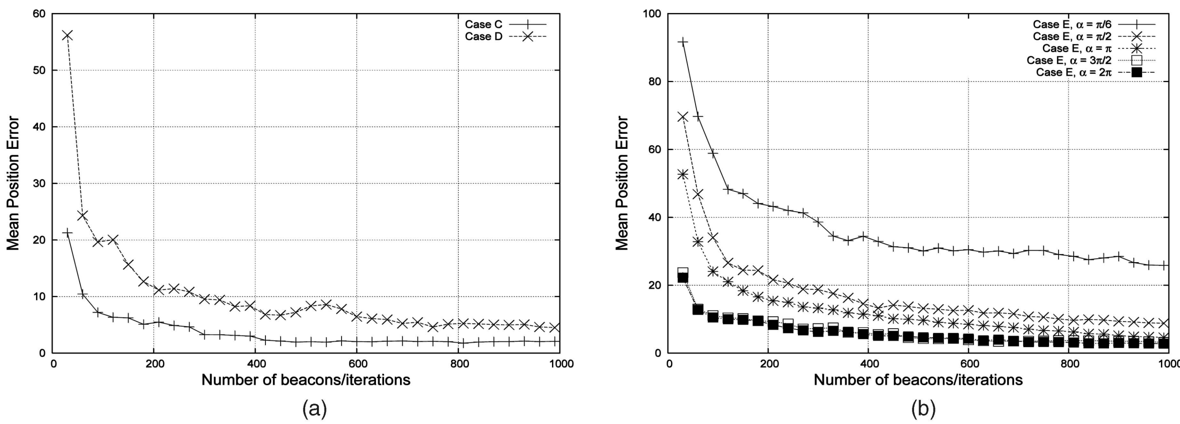


Fig. 3. The effect of GDOP on the convergency. (a) Case C versus Case D. (b) Case E: the effect of α.

and (600, 1100) are chosen as the location of two sensor nodes representing cases C and D; the initial positions for the two sensor nodes are both set as (600, 600), which is at the same distance from the two nodes; finally, we have assumed 10 percent ranging error in the RSS-based distance measurements. As is clearly shown in Fig. 3a, case C converges significantly faster than case D. The position error shown in the figure is defined as the average distance from estimated positions to true positions.

Case E in Fig. 2 could be considered as a generalized format of case C in a twofold sense:

1. to simulate the situation that a portion of the beacons are blocked by obstructions and
2. to simulate an arc-shaped moving trajectory.

Intuitively, the size of the angle α (in Fig. 2 case E) should have a nonneglectable effect on the localization performance. This has been verified through simulations shown in Fig. 3b. Interestingly, using more beacons (LA moves back and forth along the arc and broadcasts beacons) may effectively compensate for the effect of GDOP. Furthermore, as along as the angle, α is larger than $\pi/2$, the convergency is not drastically slower than the case of a full circle (2π). In real applications, the situation of a sensor node being blocked from most directions is rare, if not impossible.

Besides the situation where beacons are blocked by obstructions, a sensor node could also intentionally drop beacons to save energy. Obviously, a viable dropping strategy is to drop beacons randomly, instead of continuously dropping beacons from some specific directions. Indeed, our simulations revealed that randomly dropped beacons have a lighter influence than continuously dropped ones.

A simple but effective moving strategy for the LA could be designed based on the above observations: The LA simply follows a circle encompassing the sensor field; it can hover for several rounds if necessary. All sensor nodes could share the same initial position value—the center of the circle, which can be estimated by the LA and then broadcast to sensor nodes. Such a strategy has been used in our performance evaluations, as shown in Fig. 5.

4.5 The Cramer-Rao Lower Bound (CRLB) of Estimations

We study CRLB to investigate how well Landscape performs in location estimations. To compare with our simulation results, we consider sensors deployed over a planar area. For a single sensor node to be positioned, its position $[x_1, x_2]^T$ is our parameter. The known positions of the LA are $\mathbf{x}^b(k) = [x_1^b(k), x_2^b(k)]^T,\ k = 1, \cdots, N$. The observed distances using RSS between the LA and the

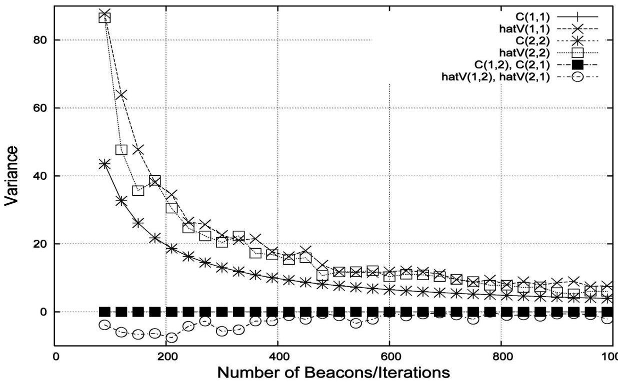


Fig. 4. Experimental results of Landscape versus CRLB.

unknown sensor are $y(k) = \xi(k) + v(k)$, where $\xi(k)$ is the true distance

$$\sqrt{(x_1 - x_1^b(k))^2 + (x_2 - x_2^b(k))^2},$$

and $v(k)$ is measurement noise, $k = 1, \cdots, N$. Now, denote $\mathbf{d} = (y(1), \cdots, y(N))^T$, $\xi = (\xi(1), \cdots, \xi(N))^T$, and $\mathbf{v} = (v(1), \cdots, v(N))^T$. We assume that $\{v(k)\}_{k=1}^N$ are independent white Gaussian noise with zero mean and variance $Var(v(k)) = \sigma^2 \xi^2(k)$, where σ^2 is a constant, $k = 1, \cdots, N$. Denote by Σ_v the variance of $\mathbf{v}$, $\Sigma_v = Diag(\sigma^2\xi^2(1), \cdots, \sigma^2\xi^2(N))$. We have the p.d.f. of $\mathbf{d}$:

$$p(\mathbf{d}) = (2\pi)^{-N/2} |\Sigma_v|^{-1/2} \exp\{-(\mathbf{d} - \xi)^T \Sigma_v^{-1} (\mathbf{d} - \xi)/2\},$$

where $|\Sigma_v|$ is the determinant of Σ_v. CRLB is the inverse of the Fisher information matrix (FIM), which is

$$J^S = E[\nabla(\ln p(\mathbf{d}))][\nabla(\ln p(\mathbf{d}))]^T.$$

We found that

$$\begin{aligned} J^S(i,i) &= (\sigma^{-2} + 2) \sum_{k=1}^{N} (x_i - x_i^b(k))^2 \xi(k)^{-4}, \quad i = 1, 2; \\ J^S(1,2) &= J^S(1,2) = (\sigma^{-2} + 2) \\ &\sum_{k=1}^{N} (x_1 - x_1^b(k))(x_2 - x_2^b(k)) \xi(k)^{-4}. \end{aligned} \tag{17}$$

We denote CRLB by C^S, which is $(J^S)^{-1}$. Now, we compare the experimental result of Landscape with the calculated CRLB. The experiment is conducted using the same configuration as the Case C shown in Fig. 3a. We denote by $\hat{V}$ the variance matrix yielded in our experiments for the estimated $[x_1, x_2]^T$. The results for $C^S(i,j)$ and $\hat{V}(i,j)$, $i, j = 1, 2$, are plotted in Fig. 4, which shows that our experimental results closely match the CRLB.

4.6 Implementation Issues

In this section, we address several practical issues related to the implementation of Landscape in real applications:

- *Beacon's transmission power.* When employing the simple moving trajectory discussed in Section 4.4, the distance from the LA to a sensor node varies and, sometimes, it could be a large value. To ensure that beacons are reachable to as many nodes as possible (so that beacons are maximally utilized), using a high transmission power to broadcast beacons may be necessary. We assume that, in real applications, the LA does not have the stringent energy constraint that a sensor node has.
- *Sensor node's initialization strategy.* In Landscape's UKF-based algorithm, the initialization of several parameters may influence the performance. Fortunately, as we will see in Section 5, Landscape allows us to use a fairly flexible initialization strategy.
- *Energy saving.* Landscape introduces zero intersensor communication overhead, which saves significant energy for sensor nodes, compared to neighborhood-measurement-based approaches. As previously discussed, a sensor node could further reduce energy consumption by intentionally dropping a portion of beacons. A sensor node could individually decide the dropping percentage based on its remaining energy and the location need (fine-grained or coarse-grained).
- *Further improvements.* Besides RSS measurement, some other ranging techniques may also be incorporated into Landscape. We discuss this issue in Section 6.

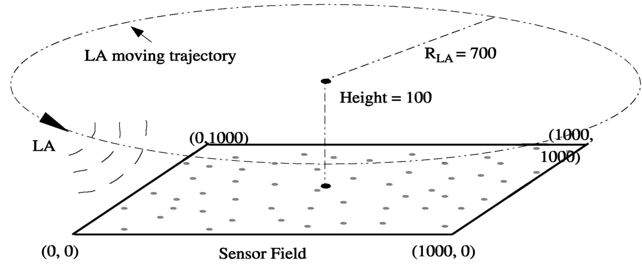


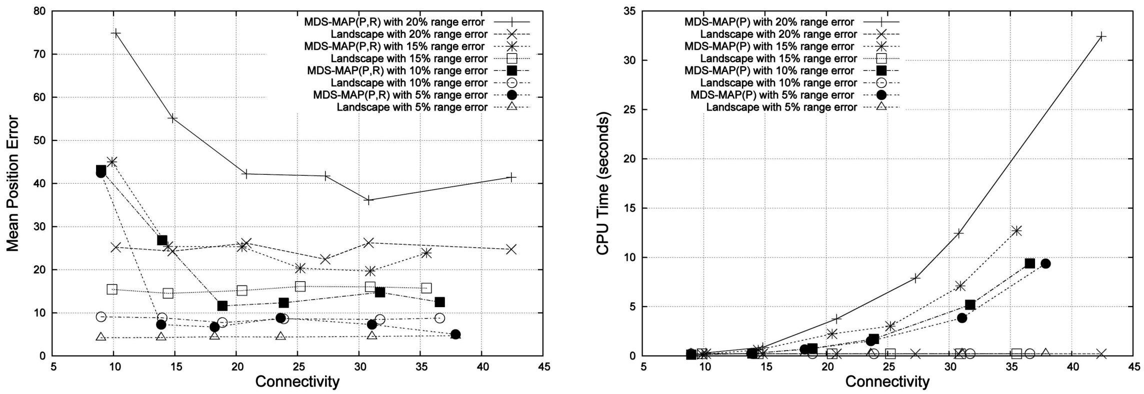
Fig. 5. Evaluation scenario.

5 Performance Evaluations

5.1 Evaluation Scenario

A simple scenario is used in our simulations. We consider a square sensor field (1,000 $\times$ 1,000) that has (0, 0), (0, 1000), (1,000, 1,000), and (1,000, 0) as its four corners. Unless explicitly specified, 200 sensor nodes are uniformly and randomly deployed in the sensor field. We let an aircraft or a balloon be the LA. As shown in Fig. 5, the LA hovers over the sensor field on a 2D plane parallel to the sensor field, moving around following a circular track with (500, 500) as the center and 700 as the radius. The height of the airplane is a constant value, for which we used 100 ft. The LA periodically broadcasts beacons to sensor nodes. Each beacon contains the transmitting power of this beacon and the LA's current location. In this scenario, the location of the LA at time step n $(n \geq 1)$ is simply

$$\begin{aligned} x_1^b(n) &= c_1 + R_{LA} \cos(2\pi/beacons\_per\_round * (n-1)), \\ x_2^b(n) &= c_2 + R_{LA} \sin(2\pi/beacons\_per\_round * (n-1)), \\ x_3^b(n) &= c_3, \end{aligned} \tag{18}$$



(a) (b)

Fig. 6. The effect of range error: Landscape versus MDS-MAP. (a) Position error. (b) CPU time.

where c_1, c_2, and c_3 are 500, 500, and 100, respectively, and R_{LA} is 700. We assume that the LA broadcasts the same number ($beacons\_per\_round$) of beacons in each round.

We assume that distance measurements are corrupted by Gaussian noise [3], [51]. A random noise is added to the true distance as follows:

$$\hat{d} = d * (1 + randn(1) * range\_error), \quad (19)$$

where d is the true distance, $\hat{d}$ is the measured distance, $range\_error$ is a value between [0, 1], and $randn(1)$ is a standard normal random variable.

5.2 Evaluation Metrics and Parameters

We code Landscape in Matlab for simulation purposes. To make our proposal comparable to other positioning schemes, we have interfaced our algorithm to the localization simulation toolkit designed as part of the Calamari project at the University of California, Berkeley [57]. We selected MDS-MAP, a state-of-the-art localization method, as a reference in the performance evaluations. Three performance metrics are generally considered for sensor localization:

- **Accuracy.** The accuracy of sensor positioning is usually presented by the average distance from estimated positions to true positions.
- **Computation overhead.** In Landscape, each sensor individually estimates its position; thus, its computation complexity is $\bigcirc(n)$ per network (n is the number of the nodes) or $\bigcirc(1)$ per node, which is same as MDS-MAP(P) [51]. However, to make the comparison more illustrative, we compared the CPU time used by these two algorithms. All simulations are conducted on a DELL Precision M50 (1.8 GHz mobile Pentium 4-M processor, 256 Mbytes DDR SDRAM) laptop with Matlab 7.0 installed. Both simulations are conducted with the Calamari simulation toolkit [57] and the execution time is averaged over each node.
- **Communication overhead.** Communications are generally more energy consuming than computations [42]. Since Landscape introduces zero intersensor communication overhead in the localization procedure, we skip this metric in our simulation results.

Since MDS-MAP(P,R) generally achieves better accuracy than MDS-MAP(P) at the cost of a higher computation overhead, we choose MDS-MAP(P,R) as the reference when evaluating the accuracy, whereas MDS-MAP(P) is used as the reference when investigating the computation overhead.

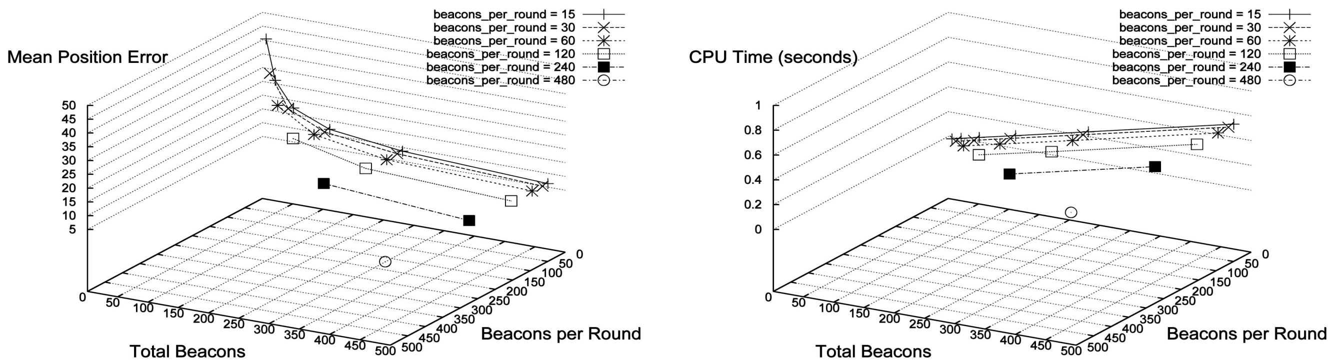
In the following sections, we vary the parameters, such as the range error, density, number of beacons, etc., to investigate their effect on performance. The influence of the initialization strategy will also be discussed.

5.3 Result versus Range Error

In this section, we investigate the performance of Landscape against various range errors. To compare with MDS-MAP(P)/MDS-MAP(P,R), whose position estimation accuracy highly depends on intersensor connectivity, we have varied connectivity values by adjusting the radio range (R) of the sensor nodes. For the simulation of Landscape, we let LA broadcast 15 beacons per round and, in total, send out 240 beacons (in 16 rounds). Ten anchor nodes were used for the simulation of MDS-MAP(P)/MDS-MAP(P,R). Experiments were conducted under four different range errors, namely, 5 percent, 10 percent, 15 percent, and 20 percent. The results are shown in Fig. 6, in which Fig. 6a compares the position error of Landscape and MDS-MAP(P,R), while Fig. 6b illustrates the CPU time used by Landscape and MDS-MAP(P). As is clearly shown in the figure, Landscape outperforms MDS-MAP in accuracy for all of the cases while, at the same time, using much less CPU time. Landscape keeps a constant value of about 0.22 sec CPU time for each node, whereas that of MDS-MAP(P,R) grows fast when connectivity is increased to enhance estimation accuracy.

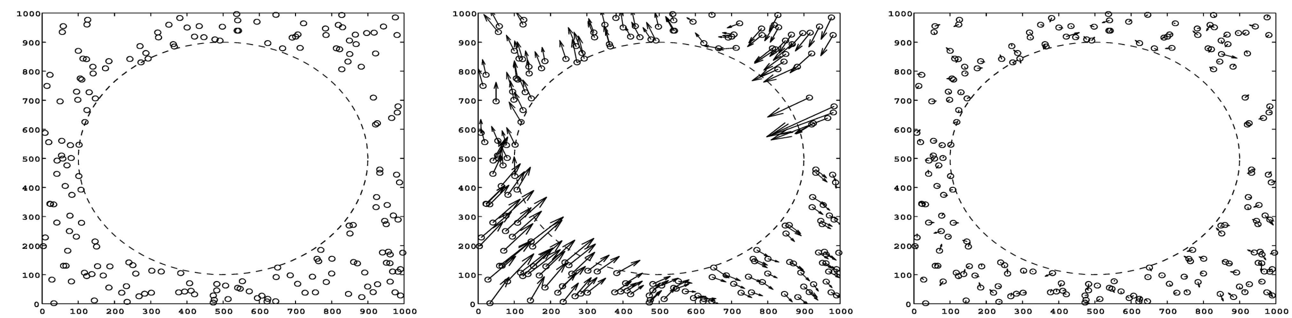
5.4 Result versus the Number of Beacons

In this section, we investigate how the number of beacons influences the performance of Landscape. As expected, the accuracy of Landscape increases with the number of beacons it uses for position estimation (the number of iterations of the UKF-based algorithm is equal to the number of beacons used). However, more beacons not only increase the cost of the LA but also introduce more



(a) (b)

Fig. 7. The effect of the number of beacons. (a) Accuracy. (b) CPU time.



(a) (b) (c)

Fig. 8. The effect of network irregularity: Landscape versus MDS-MAP. (a) Original map. (b) Result of MDS-MAP(P,R). (c) Result of Landscape.

computation overhead for each sensor node. On the other hand, if a node has only enough power to support a specific number of computation iterations (in the UKF-based algorithm), we want to know which strategy is better, a small number of $beacons\_per\_round$ with more rounds or a large number of $beacons\_per\_round$ with fewer rounds. In this group of experiments, we varied the value of both $beacons\_per\_round$ and $total\_beacons$ from 15 to 480. As is shown in Fig. 7a, with a fixed $beacons\_per\_round$ value, the positioning error decreases if we use more beacons. However, it is worthwhile noting that, after 120, the accuracy increases slowly with $total\_beacons$; with a fixed $total\_beacons$ value, however, a smaller value of $beacons\_per\_round$ yields slightly higher accuracy than a larger one. The computation cost increases linearly with $total\_beacons$, almost independent of the value of $beacons\_per\_round$, which is demonstrated in Fig. 7b. The simulation gives us helpful information from two aspects: 1) A sensor node prefers the LA to send a moderate number of beacons per round and 2) sensor nodes can individually make trade-offs between accuracy and computation cost by using a specific portion (or all) of the beacons. The range error was set as 10 percent in this group of experiments.

5.5 Result versus Network Irregularity

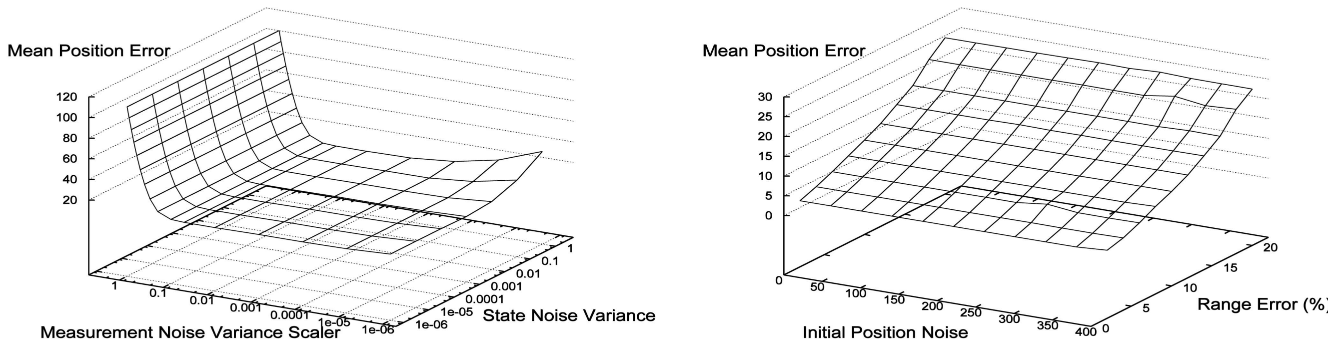
In this section, we use a simple case to demonstrate the robustness of Landscape to network irregularity. We assume that there is a lake in the middle of the sensor field. The lake is of round shape, having a radius of 400 and (500, 500) as its center. Two hundred sensor nodes are randomly deployed over the sensor field around the lake. The localization results of Landscape and MDS-MAP(P,R) are demonstrated in Fig. 8, in which Fig. 8a shows the original map of the sensors, Fig. 8b shows the result of MDS-MAP(P,R), and Fig. 8c shows the result of Landscape. In the figures, small circles represent the original location of sensor nodes, while small arrows point to the estimated positions. As is clearly shown in the figures, MDS-MAP(P,R) does not work well for the case, although the average connectivity of the network is as high as 32.97, whereas Landscape gives very good position estimations for all of the sensor nodes.

5.6 Result versus Initialization Parameters

Initialization parameters often have nonneglectable effects on the performance of KFs. For Landscape, we have the following four parameters to initialize:

1) initial position $\mathbf{x}(0)$,
2) initial position variance K_0,
3) variance matrix R for distance measurement noise,
4) variance matrix Q for state noise.

As discussed earlier, a simple strategy is to choose the center point of the LA's moving trajectory as the initial position for all sensor nodes. Therefore, we have



(a) (b)

Fig. 9. The effect of the initialization strategy. (a) Accuracy versus variance parameters. (b) Accuracy versus initial position noise.

$$\mathbf{x}(0) = (land\_size/2, land\_size/2), \tag{20}$$

$$K_0 = Diag([(land\_size/2)^2, (land\_size/2)^2]), \tag{21}$$

where $land\_size$ is the size of the sensor field. Since we are considering static sensor nodes, the state noise has small variances. Although it is rather simple to cull the parameters for Q, the initialization scheme for R could be challenging. We introduce a scaler called the $measurement\_noise\_variance\_scaler$ to investigate the influence of R on the performance of our algorithm. The expressions of R and Q are

$$R = ((R_{LA} * 2 * range\_error)^2 * measurement\_noise\_variance\_scaler), \tag{22}$$

$$Q = Diag([process\_noise\_variance, process\_noise\_variance]), \tag{23}$$

where $(R_{LA} * 2 * range\_error)^2$ is actually the upper bound of R, so the scaler takes values smaller than one. Fig. 9a shows the effects of variance parameters on accuracy. As we can see, although the value of $process\_noise\_variance$ does not have a significant impact on the results, $measurement\_noise\_variance\_scaler$ does have a nontrivial influence on the accuracy. However, as shown in Fig. 9a, as long as the scaler for measurement noise variance lies in the broad interval of [0.1, 0.00001], its impact on the accuracy is negligible. As a result, we can see that the initializations of R and Q are fairly flexible.

Unless explicitly specified, all of the simulations presented in this work are conducted using the initialization scheme quantitatively described by (20)-(23). Moreover, we have chosen 0.0001 and 0.01 for $process\_noise\_variance$ and $measurement\_noise\_variance\_scaler$, respectively.

When it is difficult to accurately locate the center point of the LA's moving trajectory, we may use an estimated center point as the initial position in our algorithm. Assuming that the estimation on the center point has Gaussian noise, the initial position will be defined as follows:

$$\hat{\mathbf{x}}(0) = (land\_size/2, land\_size/2) + randn(1, 2) * initial\_position\_noise, \tag{24}$$

where $initial\_position\_noise$ decides how large the noise is when estimating the position of the center point. Fig. 9b demonstrates the effects of $initial\_position\_noise$ on the final accuracy of Landscape. The simulations were conducted under different range errors. As we can see in Fig. 9b, compared to the influence of the range error, the impact of the initial position error on accuracy is much less significant.

6 Further Improvements Utilizing New Observations

At the current stage, Landscape aims at the applications of outdoor sensor networks. RSS measurement in an outdoor environment is usually more reliable than that in an indoor environment. Whitehouse and Culler have also shown [56] that calibrations can be used to greatly enhance the accuracy of RSS measurement. However, calibrating a large amount of sensor nodes could be expensive and time consuming. Furthermore, we should not assume that sensor networks are always deployed in a friendly environment where link qualities are good and there is no interference. The above consideration has motivated us to exploit possible mechanisms to further enhance the robustness of Landscape.

A natural and direct extension of Landscape is to incorporate other observations in Landscape's UKF-based state estimation algorithm, that is, to add a second dimension to the observation vector described in (1). We discuss two such extensions in the following sections.

6.1 Utilizing the TDoA Observation

This extension originated from the following question: Since, so far in Landscape, beacons are utilized independently, can the variance of beacons also be utilized for location estimation? Fortunately, the answer is yes and we have defined a new observation to exploit such an idea. This new observation is TDoA-based; however, it is slightly different than those TDoA measurements discussed in Section 2. We define this observation as the difference between the traveling time (from the LA to a sensor node) of two consecutive beacons. We discuss the details in the following text.

The traveling time for the $n\text{th}$ beacon observed by the $i\text{th}$ sensor node is described by the following equation:

$$t_i(n) = ((T_{ir}(n) - D_{ir}(n)) - (T_s(n) + D_s(n)),$$

where $T_s(n)$ are the timestamps of the $n\text{th}$ beacon stamped before it is sent out by the LA, $D_s(n)$ is the system delay between the time spot where the $n\text{th}$ beacon is stamped and the time spot where it is physically sent out through the antenna of the LA, $T_{ir}(n)$ is the time spot where the $n\text{th}$ beacon is received and stamped by the $i\text{th}$ sensor node, and $D_{ir}(n)$ is the system delay between the time spot when the $n\text{th}$ beacon reached the antenna of the $i\text{th}$ sensor node and the time spot where it is stamped. Similarly, the traveling time for the $(n-1)\text{th}$ beacon observed by the $i\text{th}$ sensor node is described by the following equation:

$$t_i(n-1) = ((T_{ir}(n-1) - D_{ir}(n-1)) - (T_s(n-1) + D_s(n-1)).$$

The time difference between the two traveling times for the two consecutive beacons is defined as

$$\Delta t_i(n) = (((T_{ir}(n) - D_{ir}(n)) - (T_s(n) + D_s(n))) - ((T_{ir}(n-1) - D_{ir}(n-1)) - (T_s(n-1) + D_s(n-1))).$$

The above equation can be rewritten as follows:

$$\Delta t_i(n) = ((T_{ir}(n) - T_s(n)) - (T_{ir}(n-1) - T_s(n-1))) - (D_{ir}(n) - D_{ir}(n-1)) - (D_s(n) - D_s(n-1)).$$

$D_{ir}(\cdot)$ and $D_s(\cdot)$ could be seen as constants (for each sensor node) with noise. Thus, we have

$$\Delta t_i(n) = ((T_{ir}(n) - T_s(n)) - (T_{ir}(n-1) - T_s(n-1))) + v_{it}(n),$$

where $v_{it}(n)$ is the zero-mean noise process describing the variance of system delay at the LA and sensor nodes. $\Delta t(n)$ is calculated by each sensor node. $T_s(\cdot)$ is based on the LA's local timer and $T_{ir}(\cdot)$ is based on the $i\text{th}$ sensor node's local timer.

The following observation model is used for $\Delta t_i(n)$:

$$\Delta t_i(n) = (y_i(n) - y_i(n-1))/c + v_{it}(n),$$

where c is the constant of the speed of electromagnetic radiation and $y_i(\cdot)$ is the distance observation described in (1) and (3).

One advantage of this extension is that no time synchronization (neither intersensor nor sensor to LA) is needed. However, a timestamping resolution (not accuracy) at the nanosecond level is needed. This could be achieved through some hardware support, for example, phase-locked loop (PLL)-based technologies. This extension introduces neglectable extra computation overhead (about 1 percent) but is able to effectively enhance the robustness of Landscape to ranging errors.

The preliminary results of this extension were reported in [59], in which we call this extension Landscape(T). Fig. 10 illustrates the enhancement of Landscape(T) over Landscape versus range error (for RSS-based distance measurements) and time measurement error (for TDoA measurements). In the simulations to generate this figure, we used the same configuration as that described in Section 5.1. An error model similar to (19) was used to dilute the time measurements. As shown in Fig. 10, the higher the range error, the more significant the enhancement of Landscape(T) over Landscape is, which is a favorable feature to us. Although the enhancement decreases when the time measurement error increases, Landscape(T) is still able to effectively alleviate position errors even when the time measurement error is as high as 80 percent.

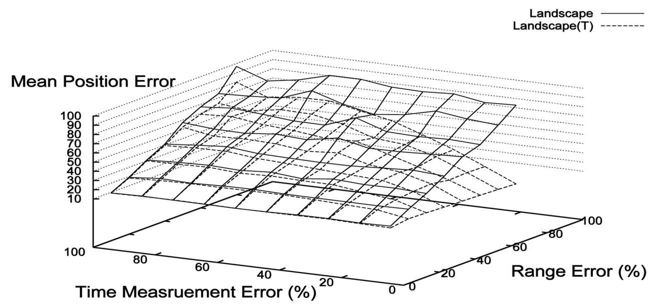


Fig. 10. The enhancement of Landscape(T) over Landscape.

6.2 Utilizing the Bearing Observation

Bearing observations, also known as angle of arrival (AoA) measurements, could also be integrated into the Landscape scheme. Like what we did with the TDoA observation, incorporating bearing observation is just adding another dimension to the observation vector. The extension is straightforward and no significant extra computation overhead will be introduced. However, to acquire bearing observations, each sensor needs to be equipped with directive antennas or antenna arrays, which may be impractical with current technologies, considering the size and cost constraints of sensor nodes. We leave this as future work for now, expecting future progress in antenna technologies will make it possible.

7 Conclusions and Future Research Directions

This paper investigates the sensor localization problem from a new prospective by treating it as a functional dual of target tracking. We propose an online estimation method called Landscape for sensor localization utilizing an LA. The state information is obtained using UKF. Landscape has several favorable features such as high scalability, no intersensor communication overhead, moderate computation cost, robustness to range errors and network connectivity, etc. The effectiveness and advantages of Landscape have been demonstrated through analysis and evaluations.

In our future research, we will extend the localization of a static sensor network to that of a slow-moving sensor network. The performance of Landscape will be studied theoretically. We shall also investigate location-aware protocols for enhanced resilience and network capacity.

Acknowledgments

This work is supported in part by an Indiana University Faculty Research Grant. The authors thank Dr. Y. Zhang and Dr. M. Fromherz at PARC and Dr. V. Raghavan at

DARPA for providing them with the MDS-MAP codes for comparisons. They also thank Dr. Y. Shang at the University of Missouri, Columbia, for providing them with information on the MDS-MAP codes.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* vol. 38, no. 4, pp. 393-422, 2002.

[2] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF Based User Location and Tracking System," *Proc. IEEE INFOCOM '00,* vol. 2, pp. 775-784, 2000.

[3] P. Biswas and Y. Ye, "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," *Proc. Third Int'l Symp. Information Processing in Sensor Networks (IPSN '04),* Apr. 2004.

[4] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Applications.* Springer, 1997.

[5] J. Bruck, J. Gao, and A. Jiang, "Localization and Routing in Sensor Networks by Local Angle Information," *Proc. ACM MobiHoc '05,* May 2005.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm. Magazine,* vol. 7, no. 5, pp. 28-34, Oct. 2000.

[7] N. Bulusu, J. Heidemann, and D. Estrin, "Density Adaptive Algorithms for Beacon Placement in Wireless Sensor Networks," *Proc. 21st IEEE Int'l Conf. Distributed Computing Systems (ICDCS '01),* Apr. 2001.

[8] W. Burgard, A. Derr, D. Fox, and A. Cremers, "Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS '98),* Oct. 1998.

[9] J.J. Caffery Jr. and G.L. Stuber, "Overview of Radiolocation in CDMA Cellular Systems," *IEEE Comm. Magazine,* pp. 38-45, 1998.

[10] J.J. Caffery Jr. and G.L. Stuber, "Subscriber Location in CDMA Cellular Networks," *IEEE Trans. Vehicular Technology,* vol. 47, no. 2, pp. 406-416, May 1998.

[11] S. Čapkun, M. Hamdi, and J.P. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," *Proc. 34th Ann. Hawaii Int'l Conf. System Sciences (HICSS '01),* Jan. 2001.

[12] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "TPS, A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[13] K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme, "Ad-Hoc Localization Using Ranging and Sectoring," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[14] L. Doherty, K. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," *Proc. IEEE INFOCOM '01,* Apr. 2001.

[15] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo Localization for Mobile Robots," *Proc. Int'l Conf. Robotics and Automation (ICRA '99),* May 1999.

[16] Enhanced 911, Fed. Comm. Commission, http://www.fcc.gov/911/enhanced/, 2007.

[17] T. Eren, D. Goldenberg, W. Whilteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[18] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM MobiCom '99,* 1999.

[19] D. Fox, J. Hightower, L. Liao, and D. Schulz, "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing,* vol. 2, no. 3, pp. 24-33, July-Sept. 2003.

[20] T. He, C. Huang, B. Blum, J.A. Stankovic, and T.F. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. ACM MobiCom '03,* Sept. 2003.

[21] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The Platforms Enabling Wireless Sensor Networks," *Comm. ACM,* vol. 47, no. 6, pp. 41-46, 2004.

[22] B. Hofmann-Wellenhoff, H. Lichtenegger, and J. Collines, *Global Positioning Systems: Theory and Practice,* fourth ed. Springer, 1997.

[23] X. Hong, K. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network Magazine,* vol. 16, no. 4, 2002.

[24] A. Howard, M. Mataric, and G. Sukhatme, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS '02),* Oct. 2002.

[25] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *Proc. ACM MobiCom '04,* Sept. 2004.

[26] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-Hoc Sensor Networks Using Multidimensional Scaling," *Proc. IEEE INFOCOM '04,* Mar. 2004.

[27] S. Julier, "A Skewed Approach to Filtering," *Proc. SPIE 12th Ann. Int'l Symp. Aerospace/Defense Sensing, Simulation, and Controls (AeroSense '98),* pp. 56-65, Apr. 1998.

[28] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proc. IEEE,* vol. 92, no. 3, Mar. 2004.

[29] C. Kwok, D. Fox, and M. Meila, "Adaptive Real-Time Particle Filters for Robot Localization," *Proc. Int'l Conf. Robotics and Automation (ICRA '03),* Sept. 2003.

[30] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM MobiCom '00,* Aug. 2000.

[31] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Proc. ACM MobiCom '98,* 1998.

[32] M. Kushwaha et al., "Sensor Node Localization Using Mobile Acoustic Beacons," *Proc. Second IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS '05),* Nov. 2005.

[33] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The Unscented Particle Filter," Technical Report CUED/F-INFENG/TR 380, Eng. Dept., Cambridge Univ., Aug. 2000.

[34] D. Moore, J. Leonard, D. Rus, and S.J. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," *Proc. Second ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* Nov. 2004.

[35] R. Nagpal, "Organizing a Global Coordinate System from Local Information on an Amorphous Computer," A.I. Memo 1666, AI Laboratory, Massachusetts Inst. of Technology, Aug. 1999.

[36] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02),* Sept. 2002.

[37] D. Niculescu and B. Nath, "Ad-Hoc Positioning System (APS)," *Proc. IEEE Global Telecomm. Conf. (Globecom '01),* Nov. 2001.

[38] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AoA," *Proc. IEEE INFOCOM '03,* Apr. 2003.

[39] P. Pathirana, A. Savkin, S. Jha, and N. Bulusu, "Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks," *IEEE Trans. Mobile Computing,* vol. 4, no. 3, May/June 2005.

[40] N. Patwari and A. Hero III, "Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks," *Proc. Second ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '03),* Sept. 2003.

[41] N. Patwari, A. Hero III, M. Perkins, N. Correal, and R. O'Dea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. Signal Processing,* special issue on signal processing in networking, vol. 51, no. 8, pp. 2137-2148, Aug. 2003.

[42] G. Pottie and W. Kaiser, "Embedding the Internet: Wireless Integrated Network Sensors," *Comm. ACM,* vol. 43, no. 5, pp. 51-58, May 2000.

[43] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. ACM MobiCom '00,* Aug. 2000.

[44] N.B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks," LCS Technical Report 892, Massachusetts Inst. of Technology, Apr. 2003.

[45] N.B. Priyantha, A. Miu, H. Balackrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," *Proc. ACM MobiCom '01,* July 2001.

[46] N.B. Priyantha, H. Balackrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," *Proc. IEEE INFOCOM '05,* Mar. 2005.

[47] A. Savvides, C.-C. Han, and M.B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM MobiCom '01,* July 2001.

[48] A. Savvides, H. Park, and M. Srivastava, "The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02),* Sept. 2002.

[49] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M.B. Srivastava, "On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks," *Proc. Second Int'l Workshop Information Processing in Sensor Networks (IPSN '03),* Apr. 2003.

[50] J. Saxe, "Embeddability of Weighted Graphs in K-Space Is Strongly NP-Hard," *Proc. 17th Allerton Conf. Comm., Controls, and Computing,* pp. 480-489, 1979.

[51] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 15, no. 11, pp. 961-974, Nov. 2004.

[52] M. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," *Proc. First IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems (MASS '04),* Oct. 2004.

[53] N. Sundaram and P. Ramanathan, "Connectivity Based Location Estimation Scheme for Wireless Ad Hoc Networks," *Proc. IEEE Global Telecomm. Conf. (Globecom '02),* vol. 1, pp. 143-147, Nov. 2002.

[54] VOR Navigation, http://www.navfltsm.addr.com/vor-nav.htm, 2007.

[55] K. Whitehouse, "The Design of Calamari: An Ad-Hoc Localization System for Sensor Network," master's thesis, Univ. of California, Berkeley, 2002.

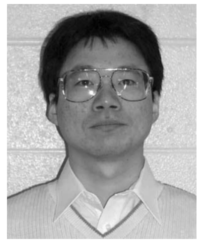
[56] K. Whitehouse and D. Culler, "Calibration as Parameter Estimation in Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02),* Sept. 2002.

[57] K. Whitehouse, "Calamari: A Sensor Field Localization System," http://www.cs.berkeley.edu/~kamin/calamari, 2007.

[58] L. Zhang, Q. Cheng, Y. Wang, and S. Zeadally, "Landscape: A High Performance Distributed Positioning Scheme for Outdoor Sensor Networks," *Proc. First IEEE Int'l Conf. Wireless and Mobile Computing, Networking and Comm. (WiMob '05),* Aug. 2005.

[59] L. Zhang and Q. Cheng, "Landscape(T): A Robust and Low-Cost Sensor Positioning System Using the Dual of Target Tracking," *Poster Proc. IEEE/ACM Int'l Conf. Distributed Computing in Sensor Systems (DCOSS '06),* June 2006.

Liqiang Zhang received the bachelor's degree in computer science from the China University of Mining and Technology, the master's degree in electrical engineering from the China Academy of Telecommunications Technology, and the PhD degree from the Department of Computer Science at Wayne State University, Detroit, Michigan. He is an assistant professor in the Department of Computer and Information Sciences at Indiana University South Bend. His research interests include wireless networks, mobile computing, quality of service, sensor networks, and network security. He is the founding program cochair of the International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN). He is a guest coeditor of the special issue on wireless mesh networks of Elsevier's *Computer Communications* journal. He is a member of the IEEE and the ACM.



Qiang Cheng received the BS and MS degrees from Peking (Beijing) University, China, and the PhD degree from the Department of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign (UIUC). During the summer of 2000, he worked at the IBM T.J. Watson Research Center, Yorktown Heights, New York. In June 2002, he joined the Electrical and Computer Engineering Department at Wayne State University, Detroit, Michigan, as an assistant professor. From May to August 2005, he was an AFOSR Faculty Fellow at the Air Force Research Laboratory, Wright-Patterson, Ohio. Since August 2005, he has been a senior researcher/research scientist at Siemens Medical Solutions, Siemens Corp., Princeton, New Jersey. His research interests include signal, image, and multimedia processing, learning theory, information theory, and various applications to multimedia computing and communications, biomedical, and healthcare. He is the recipient of various awards, scholarships, and privileges, for example, he received the Achievements Awards from Siemens Corp. from 2006 to 2007, the University Research Awards from Wayne State University in 2003, and the Invention Achievement Award from the IBM T.J. Watson Research Center in January 2001. He held a University Fellowship at UIUC from 1997 to 1998 and a Guanghua University Fellowship at Peking University, China, from 1995 to 1996. He received the Antai Award for Academic Excellence from Peking University, China, in 1995. He is a member of the IEEE.

Yingge Wang received the BS degree from Peking University, China, in 1996 and the MS degree from the Electrical and Computer Engineering Department at Wayne State University, Detroit, Michigan. His research interests include signal processing, biomedical information processing, and pattern recognition.

Sherali Zeadally received the BA and MA degrees in computer science from the University of Cambridge, England, and the doctorate degree in computer science from the University of Buckingham, England, in 1996. He is currently an associate professor in the Department of Computer Science and Information Technology at the University of the District of Columbia, Washington, D.C., where he directs the Network Systems Laboratory. He currently serves on the editorial boards of several international journals and he is a co-editor-in-chief of the *International Journal of Internet Protocol Technology.* His research interests include computer networks (wired and wireless), mobile computing, network and system security, ubiquitous computing, multimedia networking, and performance evaluation of networks and systems. He is a senior member of the IEEE, a fellow of the British Computer Society (FBCS), and a fellow of the Institution of Electrical Engineers (FIEE), United Kingdom.