Abstract: In this work, we consider the sensor localization 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/speed of a moving target. As a dual, we utilize a moving location-assistant (LA) (with global positioning system (GPS) or pre-defined 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 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 received signal strength (RSS), and individually calculates their locations via an Unscented Kalman Filter (UKF) based algorithm. Our contributions are at least twofold. (1) Landscape is a distributed scheme, it does not rely on measured distances among neighbors (as used by most current proposals), which makes it robust to topology and density; Landscape involves zero sensor-to-sensor communication overhead, and is highly scalable to network size. (2) By introducing UKF in sensor localization problem, we reap multiple benefits: our UKF-based algorithm nicely exploits the constraints increasingly added by the beacons; it elegantly solves the nonlinear problem with low computation cost and complexity; and most importantly, it efficiently reduces the effects of measurement errors, making Landscape robust to ranging errors. Extensive simulations and evaluations against the state-of-the-art systems show that Landscape is a high-performance sensor positioning scheme for outdoor sensor networks.

Keywords: Wireless Sensor Networks; Localization algorithm; Unscented Kalman Filter

1 Introduction

Recent years have witnessed explosive interest in wireless sensor networks (WSNs) both from industries and academia. It is believed that WSNs will revolutionize the way we understand and construct complex physical systems [17], and extend our sensory capability to every corner of the world [11]. 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. Examples of those applications may include battlefield surveillance, intrusion detection, forest fire detection, smart environment, and others [1]. In most of these 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 discounted. Location-aware sensors may also help to highly enhance the efficiency of routing protocols [30, 31, 32, 52] by reducing costly message flooding. However, equipping a global positioning system (GPS) [21] receiver to each sensor node may not be a practical solution for most applications, because of the constraints in size and cost of construction of sensor networks.

Localization has been studied for many years as a classical problem in many disciplines including the navigation systems (VOR [53]) and GPS [21]), the robot localization problem in mobile robotics [14, 23, 29], user location
identifying in cellular networks [15, 8, 9] and WLANs [2, 41]. However, solutions for the above problems may not be directly applicable to networked sensors. Recently, Location discovery of WSNs has attracted extensive research efforts. It remains a challenging problem, however, because several characteristics are highly desirable for sensor localization schemes [11]. (1) The positioning algorithm should be distributed in order to scale well for large sensor networks. (2) The localization protocol must minimize 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 pre-set location) or neighbors. When the percentage of anchor nodes (among total nodes) is high enough that each node has three anchor nodes (non-collinear) 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 which assume only a small fraction of anchor nodes [3, 5, 6, 13, 25, 45, 46] or even anchor-free [19, 42]. 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 geometric graph embedding problem [16, 42, 48, 57], or more generally, a constrained optimization problem [3, 13, 46]. However, to obtain 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 [13] may involve high computation complexity becoming usable only when the application permits deployment of a central processor to perform location estimation [39]; on the other hand, some distributed solutions (such as AFL [42], which uses mass-spring model for refinement) may be not able to avoid high volume interactive communications among neighbors. Furthermore, localization methods based on measurements of neighborhood begin to perform acceptably only at node densities well beyond the density required for network connectivity [12].

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, could be aircraft, balloon, robot, 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 a single or multiple static location-aware sensors to track and predict the position (and/or speed) of a moving target. In our proposed system, we let each location-unaware sensor discover its position by passively observing a moving, location-aware LA (with the GPS or pre-defined moving path). We resolve this functional dual problem by modeling and utilizing an Unscented Kalman Filter (UKF) [28] based algorithm. One scenario involving sensor networks frequently mentioned in the literature is that of aircraft deployment of sensors followed by in flight collection of data by cruising the sensor field [37]. Landscape fits well (however, not limited) for 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 methods applicable for outdoor sensor networks:

- **Landscape is fully-distributed and localized making it highly scalable.** Each node discovers its location with its own measurements and calculations. There is no inter-sensor communications and dependencies. It is applicable to large areas of sensor networks with arbitrary densities.

- **LA broadcasts the beacons and the sensor nodes listen passively.** There is no interactive sensor-to-sensor communication.
communications involved in this process, which not only saves the sensor’s energy, but also lessens the channel congestion. Approaches based on the neighborhood measurements cannot totally avoid message flooding [5, 49, 50].

- **Landscape relies on radio-frequency (RF) RSS to measure distance.** It does not require the sensor node to be equipped with any ultrasound receiver or antenna array which is needed for time difference of arrival (TDoA) or angle of arrival (AoA) measurements. It does not have any synchronization requirement on the network, which may be needed for Time of Arrival (ToA) measurement.

- **Landscape uses UKF-based algorithms which introduce only moderate computational cost**, compared to other sequential techniques such as Monte Carlo based methods [14, 25, 29]. Landscape provides cost-accuracy flexibilities. Each sensor node may use less beacons (less computations) for a lower accuracy, or vice versa. The details will be given in the following sections.

- **Landscape is robust to range-errors.** Our simulations show that it gives satisfactory result with low computational cost (and improves further with more beacons) even when range errors is up to 20%.

- **Landscape is suitable to arbitrary network topologies**, and can be used for sensor networks deployed in complex outdoor environments, thanks to the absence of sensor-to-sensor connectivity requirement.

Extensive simulations have been conducted to study the performance of Landscape. We have used MDS-MAP [49, 50, 51] as a reference for evaluation purposes, because MDS-MAP represents a state-of-the-art approach to sensor localization based on neighborhood measurements. Extensive simulations reveal that Landscape gives better results than MDS-MAP with less computation and communications cost.

The rest of this paper is organized as follows: the next section presents the related work. We describe our proposed approach in Section 3 and 4. Performance evaluations of our proposed system are presented and discussed in Section 5. Section 6 presents our concluding remarks.

2 Related Work

Sensor localization has attracted significant research effort in recent years, and various approaches have been proposed. The majority of them assume that a small fraction of the nodes (called anchors or beacons) have a priori knowledge of their locations. Most of them also follow a common process for location discovery: The first phase is to make the estimation of distances or angles to anchors or other neighboring nodes, which is often called ranging. The second phase is to estimate positions based on the ranging measurements. Some proposals have an optional third phase, which is to refine the position estimations utilizing the local [46, 50] or global information [49]. 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 (ToA/TDoA, AoA, and RSS); then we use a separate subsection to describe MDS-MAP related algorithms [49, 50, 51]; finally, we take a glance at Bayesian techniques for robot location estimations.

2.1 Location Discovery with T(D)oA Measurement

RF time of arrival (ToA) is a common technology used to measure distance via signal propagation time. Since RF signal travels 0.3 meters per nanosecond, 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 [39]. Multipath and NLOS (non line of sight) are two other sources of errors for ToA measurements. The most basic localization system using ToA is the GPS [21], which requires relatively expensive and energy-consuming electronics to precisely
synchronize with a satellite’s clock [19].

Time difference of arrival (TDoA) uses the difference between the traveling times of two signals to estimate distance. For example, in AHLoS [45], 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 (e.g. ultrasound) travels at a much slower speed making it not as sensitive to timing as ToA. However, equipping ultrasound to sensor node not only means more cost and energy consumption but also requires nodes to be densely deployed (ultrasound usually only propagate 20-30 feet), which makes it unsuitable for outdoor applications.

Čapkun et al. [10] 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. Each node builds a relative coordinate system, in which the node itself is the origin. Each node thus calculates the coordinates of all its one-hop neighbors using triangulation. These local coordinate systems are then merged together to form a network coordinate system. The process of merging local maps may involve error propagations or accumulations, and introduces more communications. This approach gives a relative map by eliminating the need of GPS-equipped anchors, and is often categorized as an anchor-free algorithm.

AHLoS [45] is a well-known TDoA-based sensor localization system, in which nodes estimate their locations by iterative multilateration and collaborative multilateration. In integrative multilateration, a sensor becomes a base station after its position is determined. Whenever a sensor has range estimates to at least three base stations, multilateration is used to compute its position; otherwise, if a sensor cannot find three base stations, collaborative multilateration is used. In collaborative multilateration, two or more sensors (which can be multiple hops apart) can form an overdetermined system of equations with a unique solution set. N-hop multilateration has been further investigated in [46, 47].

Cheng et al. proposed TPS in [11], a novel time-based positioning scheme for outdoor sensor networks, which utilizes 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. TPS employs three location-aware base stations \(A\), \(B\), and \(C\) placed beyond the boundary of the monitored sensor field. The master station \(A\) broadcasts a beacon every \(T\) seconds. Station \(B\) replies a beacon to \(A\) after receiving the beacon from \(A\), and \(C\) replies a beacon to \(A\) after receiving both beacons from \(A\) and \(C\). Each node uses TDoA to measure the time difference among the three beacons arrival from \(A\), \(B\) and \(C\) and get a set of equations. Its location can then be calculated by resolving the equations. TPS shares several advantages with our approach, e.g., fully-distributed and involving zero communication overhead. However, TPS has an extremely high requirement on the accuracy of time measurements (at the level of nanoseconds), which can be impractical for low-end sensor nodes.

### 2.2 Location Discovery with AoA Measurement

In angle of arrival (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 [53] 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 [9]; (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 [37]. To avoid using directive antenna 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 Cricket Compass project [43] at MIT. Two
algorithms were examined by Niculescu and Nath: \textit{DV-Bearing}, which allows each node to get a bearing to a landmark, and \textit{DV-Radial}, which allows a node to get a bearing and a radial to the landmark. A node's location can be calculated immediately if it has two anchors (for \textit{DV-radial}) or three anchors (for \textit{DV-Bearing}) in its neighborhood. The authors investigated orientation forwarding to propagate the measurements when a node does not have enough anchors in its neighborhood. One advantage of this approach is that it provides orientation besides coordinates. However, to get good performance, error control mechanisms have to be carefully designed to reduce the error propagation involved during orientation forwarding [37].

A directionality based scheme was proposed by Nasipuri and Li in [35]. They transferred the need for directive antennas from the sensors to anchors. In their system, four fixed stations are placed on the corners of the sensor network area. Each station sends beacon signals consisting of continuous RF carrier signals on a narrow directional beam that rotates with a constant angular speed. The four stations are wired and controlled by a central controller so that they are synchronized to keep a constant angular separation between the directional beams from them. Each node records the time instants when it receives the different beacon signals, and evaluates its angular bearings and location with respect to the beacon stations by triangulation. This proposed system requires strict line of sight (LOS) between each station and each sensor. Non-zero antenna beam width and imperfect synchronization may lead to decreased system accuracy [11].

\subsection{2.3 Location Discovery with RSS Measurement}

Received signal strength (RSS) has been widely used as a distance measure in the context of WSNs because of its simplicity. In RSS method, the measured received power and the known transmit 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 [38, 44]. RSS-based localization has been studied extensively [2, 3, 26, 36, 38, 39, 41, 54]. An extreme case of RSS method is connectivity (or proximity) measurement, in which only the information of whether or not two devices are connected or “in-range” is used. The schemes [5, 13, 19, 25, 34, 49], based on connectivity, are often referenced as range-free algorithms, which provide coarse-grained localization only but show robustness in situations where RSS is hard to estimate accurately such as complex indoor or urban environments.

Ad hoc positioning system (APS) [36] first estimates ranges based on DV-hop, DV-distance, or Euclidean, then applies trilateration to compute the location of each sensor. If enough base stations are available, location errors for APS with DV-hop can be about 30\% of the radio range in a dense network with a regular topology. For sparse and irregular networks, the accuracy degrades to roughly the radio range. For DV-distance and Euclidean, the performance of APS also depends on the accuracy of the distance measured between neighboring nodes [11].

Bulusu et al. [5] 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. [19] proposed another range-free positioning scheme called APIT. The possible area in which a target node resides is narrowed down by a PIT (point in triangulation) test. In this test, a node chooses three anchors from all audible anchors and tests whether it is inside the triangle formed by connecting these three anchors. APIT repeats this PIT test with different audible anchor combinations until all combinations are exhausted or the required accuracy is achieved. APIT calculates the center of gravity of the intersection of all of the triangles in which a node resides to determine its estimated position.

Patwari and Hero [38, 39] studied the Cramer-Rao Bound (CRB) for location estimation based on RSS measurement. Whitehouse and Culler [54, 55] designed and evaluated an ad-hoc localization system called Calamari. They have offered 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 mea-
suring RSS, e.g. outdoors [54]. Furthermore, Whitehouse and Culler have shown [55] that calibration can be used to highly enhance the accuracy of RSS measurement.

2.4 Multidimensional Scaling for Localization

Multidimensional scaling (MDS) [4] has recently been successfully used to resolve sensor localization problem [49, 50, 51, 26]. Originated in 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]. MDS nicely matches the sensor positioning problem in a nature way. One main advantage in using MDS is that it can always generate relatively high accurate position estimation even based on limited and error-prone distance information [26]. Shang et al. first proposed MDS-MAP to use MDS in sensor location problem in [49]. MDS-MAP is a centralized algorithm, which consists of three steps:

1. Compute 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 2 (or 3) largest eigenvalues and eigenvectors to construct a 2D (or 3D) relative map.

3. Given sufficient anchors (3 or more for 2D, 4 or more for 3D), transform the relative map to an absolute map based on the absolute positions of anchors.

MDS-MAP(P) [50] is an improved 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-hops neighbors) and then the local maps are merged to form a global map. If an optional refinement process is used for each local map before merging, the algorithm is called MDS-MAP(P,R).

MDS-MAP(P,R) has been shown to have very impressive performance [50]. For a random uniform sensor network with 6 anchors and connectivity equal to 20, when the proximity is used, the median location error of MDS-MAP(P,R) is less than 20% of the radio range; and when the range is used with 5% error, the median location error could be less than 5%.

2.5 Bayesian Techniques for Robot Localization

Bayesian techniques have been widely investigated in the context of robot localization [18, 23]. Recently, grid based Markov localization [7], particle filter (a.k.a. sequential Monte Carlo) [14], 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 range, bearing 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 MCL (Monte Carlo Localization) [25]. Inspired by the techniques used for robot localization, they first proposed to use sequential Monte Carlo (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 mainly for static sensor networks. (2) MCL utilizes only proximity measurement, with the location estimation coarse-grained and bounded. In contrast, Landscape exploits range measurement and is able to acquire high accuracy. (3) With SMC requiring intensive computation power, upgrading MCL for range measurements might be impractical, because that would highly increase the computation
cost of MCL. To the best of our knowledge, we are the first to utilize UKF [27] in location estimation, which is
easy to implement and have high accuracy in the presence of nonlinear observation functions and nonGaussian
distributions. The approximations to the nonlinear functions can be accurate to the second-order Taylor expansion
for arbitrary distributions, while the computations are well controlled and comparable to linearization.

3 Landscape Localization Methodology and Model

3.1 Landscape Methodology

In this paper, we design our sensor localization system Landscape with a location assistant (LA), e.g., 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 problems. In target tracking, one (or more)
location-aware sensor node estimates the position (and optionally, velocity and acceleration) of a moving target
based on the measurable distances or AOAs. As a functional dual, each location-unaware sensor node utilizes
the measured RSS to estimate its own position aided by the location-aware LA. From this novel perspective, our Land-
scape system exploits 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 the 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.

3.2 Landscape Localization Model

For the localization described above, we define the state variable as the (unknown) 3-D position of a specific sensor
node,

\[ \mathbf{x}(n) = \{x_1(n), x_2(n), x_3(n)\} . \]  

And we have the following dynamic state and observation equations:

\[ \begin{align*}
\mathbf{x}(n) &= \mathbf{f}(\mathbf{x}(n-1)) + \mathbf{w}(n), \\
\mathbf{y}(n) &= \mathbf{g}(\mathbf{x}(n)) + \mathbf{v}(n),
\end{align*} \]  

where \( \mathbf{f}(\cdot) \) and \( \mathbf{g}(\cdot) \), respectively, are state evolvement and observation functions. \( \mathbf{f}(\cdot) \) may be linear or nonlinear
depending on application scenarios, while \( \mathbf{g}(\cdot) \) is usually highly nonlinear. \( \mathbf{w}(n) \) and \( \mathbf{v}(n) \) are state and observa-
tion noise sequences.

One interesting application considers static sensor localization, where the positions of the sensors remain un-
changed after deployment. That is, the state dynamics \( \mathbf{f}(\cdot) \) governing the sensor positions are simply the identity
functions:

\[ \mathbf{x}_i(n + 1) = \mathbf{x}_i(n) + \mathbf{w}_i(n), \quad 1 \leq i \leq 3. \]  

with \( \mathbf{w}_i(n) \) modeling the small position perturbation due to the wind or other environmental effects. Our algo-
rithm can 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, which can be delivered to sensor nodes.
Equipped with accurate GPS, the LA knows its current location. The current position can be transmitted through
RF signal to the ground sensors. The following observation model is commonly used in practice:

\[ y(n) = \sqrt{(\Delta x(n))^2 + (\Delta y(n))^2 + (\Delta z(n))^2} + v(n). \]
Here \( \Delta x(n) = x^t(n) - x_1(n) \), \( \Delta y(n) = y^t(n) - x_2(n) \), \( \Delta z(n) = z^t(n) - x_3(n) \). \((x^t(n), y^t(n), z^t(n))\) is the current 3-D position of the LA, measured using GPS or controlled by the pre-defined path. And \( v(n) \) models the observation error, which usually comes from the RF distance estimations or the perturbations to the LA positions. We assume \( \mathbf{w}(n) \) and \( v(n) \) are zero-mean uncorrelated noise processes.

In general, as long as the LA knows its own instant position and broadcasts this information to the sensors, each sensor can use our algorithm to locate itself with the on-line estimations. The estimation improves with accumulated observations and recursive update. No inter-sensor or sensor-to-LA communications is needed. Specifically, we require the LA to broadcast RF messages to the sensors with constant time intervals, in which the current LA position \((x^t(n), y^t(n), z^t(n))\) is contained,

\[
[x^t(n), y^t(n), z^t(n)]^T = h(n\Delta T; x, y, z),
\]

where \( \Delta T \) is the time interval of broadcasting, and \( h(\cdot) \) is the locus of the LA. In practice, we often encounter those applications with a sensor network deployed approximately on a plane. In this case, we are mainly interested in estimating the planar sensor positions. We design the LA to move on a 2-D plane parallel to the sensors, then \( c_z - x_3(n) \overset{\Delta}{=} \Delta h \). This design is useful for the planar sensors, and reduces the estimation from 3-D to 2-D. The third equation in the state process (3) may be ignored. In Landscape, we apply the unscented Kalman filtering (UKF) which is an elegant and computationally efficient recursive state estimation technique and well suited to our applications.

4 Landscape State Estimation Via Unscented Kalman Filtering

Our Landscape localization system aims at improving the sensor localization by iteratively updating the position estimates with the current observations. For the system models defined in the previous sections, on-line state estimation need be performed. Kalman filters and their variants have been designed for this purpose, but their actual performance depends heavily 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 rooted-sum-of-squares of position difference, standard Kalman filter (KF) is not suitable to our applications. Neither is the extended Kalman filter (EKF), the first-order approximation to the nonlinear system that is often plagued by the empirical linearization. For the nonlinear observation function \( g(\cdot) \), the unscented transformation (UT) [33] [27] is an elegant approach to providing higher-order approximations. It can accurately compute the statistical mean and variance up to the third-order of Taylor series expansion of \( g(\cdot) \) for Gaussian noise sequences, or the second-order for arbitrary noise distributions. Higher-order approximation can also be captured with extended algorithms [28]. At the same time, UT uses the same order of calculations as linearization. We shall apply the unscented Kalman filter (UKF) [27] in our Landscape system.

4.1 Unscented Transformation

The UT has been developed to handle low-order statistics of random variables that undergo a nonlinear transform \( g(x) \). The knowledge of higher order information can also be partially incorporated into the sigma point set. Let \( D_x \) be the dimension of \( x \), \( \mu_x \) be the mean, and \( K_x \) be the variance matrix, the UT calculates the first two moments in the following way:

1. Generate a set of sigma points \( S = \{x_k, W_k : k = 0, \ldots, 2D_x \} \), with \( W_0 \) denoting the weight on the mean point:

\[
\begin{align*}
\mathbf{x}_k &= \mu_x + \left( \sqrt{\frac{D_x}{1 - W_0}} \right) K_x^k, & W_k &= \frac{1 - W_0}{2D_x}, \\
\mathbf{x}_{k+D_x} &= \mu_x - \left( \sqrt{\frac{D_x}{1 - W_0}} \right) K_x^k, & W_{k+D_x} &= \frac{1 - W_0}{2D_x},
\end{align*}
\]

where
for $k = 1, \cdots, D_x$.

2. Propagate the sigma points through the nonlinear transformation

$$y_k = g(x_k), \quad k = 0, \cdots, 2D_x.$$  \hfill (7)

3. Calculate the mean and variance of the transformed points

$$\mu_y = \sum_{k=0}^{2D_x} W_k y_k, \quad K_y = \sum_{k=0}^{2D_x} W_k (y_k - \mu_y)(y_k - \mu_y)^T. \hfill (8)$$

The UT can capture the first two statistical moments up to the second-order of the Taylor series.

### 4.2 Unscented Kalman Filter

The unscented Kalman filter (UKF) embeds the UT into the KF’s recursive prediction and update structure. The general formulation expands the state vector with the process and observation noise $x^a(n) = [x^T(n), w^T(n), v^T(n)]^T$. The resultant augmented vector is of dimension $D_a = D_x + D_w + D_v$. The process and observation models for the augmented vector $x^a(n)$ from (2) are

$$x^a(n) = f^a(x^a(n-1)), \hfill (9)$$
$$y^a(n) = g^a(x^a(n)).$$

The UKF is implemented as follows [27]:

1. Initialization:

$$x^a(0) = [x^T(0), 0, 0]^T, K^a(0) = \text{Diag}(K_0, Q, R), \hfill (10)$$

where $Q$ and $R$, respectively, are the variances for noise processes $w(n)$ and $v(n)$.

2. Iteration for $n$:

(a) Applying sigma points procedure to the augmented system (9), and the resulting sigma points are $\{x^a_k(n), W_k : k = 0, \cdots, 2D_a \}$.

(b) Prediction:

$$x^a_k(n) = f^a(x^a(n-1)), \hfill (11)$$
$$\mu^a_x(n) = \sum_{k=0}^{2D_a} W_k x^a(n), \hfill (12)$$
$$K^a(n) = \sum_{k=0}^{2D_a} W_k (x^a(n) - \mu^a(n)) \times (x^a(n) - \mu^a(n))^T, \hfill (13)$$
$$y(n) = g^a(x^a(n)), \hfill (14)$$
$$\mu_y(n) = \sum_{k=0}^{2D_a} W_k y(n). \hfill (15)$$
With the same order of calculations as the EKF, the UKF can approximate the second-order Taylor series expansion for arbitrary distributions. In contrast, the EKF can only approximate the first order; therefore, the UKF is much more accurate, as demonstrated by many applications [27] [33]. Compared to particle filtering (PF), the UKF uses a deterministic set of sigma points instead of a large number of particles. Thus, the UKF can be implemented in a well controlled manner. Because of its implementation simplicity and high accuracy, we exploit the UKF for state estimation in our Landscape localization system.

4.3 Landscape Localization Via UKF

With the system model described in Section 3, the range of the LA is measured. If the bearing can be also measured, we expect a higher degree of accuracy, but with more equipment and computations. As shown by our simulation results, the accuracy is already sufficiently high with range information alone. And the accuracy may as well improve with more iterations of UKF. Therefore, our Landscape system only makes use of RF range information. The inclusion of the bearing information is straightforward.

When the LA updates its positions with a moderately high frequency, a short period of time may be sufficient for satisfactory localization accuracy. For example, in our simulation, several rounds of hovering of an airplane work well with more than 10 updates per round, with the accuracy already higher than the MDS as shown by our evaluation results. And the accuracy improves with more rounds.

5 Performance Evaluation

5.1 Evaluation Scenario

As presented in previous sections, Landscape does not rely on how sensor nodes are distributed and does not have specific requirements on the LA’s moving trajectory. However, for simulation purpose, we have selected a simple scenario. We use a square sensor field (1000 by 1000) with (0,0), (0, 1000), (1000,1000), and (1000, 0) as the 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 Figure 1, the LA hovers over the sensor field on a 2-D plane parallel to the sensor field, moving around following a circle 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 feet here. The LA periodically broadcasts beacon samples to sensor nodes. Each beacon sample 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:

\[
\begin{align*}
    x'(n) &= c_x + R_{LA} \cos(2\pi/samples\_per\_round \times (n - 1)), \\
    y'(n) &= c_y + R_{LA} \sin(2\pi/samples\_per\_round \times (n - 1)), \\
    z'(n) &= c_z.
\end{align*}
\]

where $c_x$, $c_y$, and $c_z$ are 500, 500, and 100 respectively, and $R_{LA}$ is 700. We assume that the LA broadcasts same number ($samples\_per\_round$) of beacon samples in each round. If real time location estimation is preferred, the moving trajectory of LA can be delivered to each sensor node as we discussed in the previous section. However, if no real time estimation is needed, sensor nodes can simply collect all the beacons (thus the moving trajectory is acquired) and then make the estimation utilizing the UKF-based algorithm. In the evaluation scenario, we assume that the LA has a large radio range, so that the beacons can be utilized to a maximum by the sensor nodes. This is a reasonable assumption, however, since the LA does not have the energy constraint as for sensor nodes. No specific requirement is placed on sensor’s radio range.

We assume distance measurements have Gaussian noise [3, 49, 50]. A random noise is added to the true distance as following:

\[
\hat{d} = d \times (1 + \text{randn}(1) \times \text{range}\_\text{error})
\]

where $d$ is the true distance, and $\hat{d}$ is the measured distance, $\text{range}\_\text{error}$ is a value between [0,1], and $\text{randn}(1)$ 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 Berkeley’s Calamari project [56]. We selected MDS-MAP as a reference for performance evaluation, since to the best of our knowledge, MDS-MAP is one of algorithms which gives the best overall performance among other neighborhood-measurement based approaches. Three performance metrics are generally considered for sensor localization:

- **Accuracy**: The accuracy of sensor positioning is usually presented by the average distance between estimated position to true position. For neighborhood-measurement based approaches (including MDS-MAP), this value is often normalized to the radio range of sensor node. Since Landscape does not depend on network connectivity (average number of neighbors), and sensor node may hold different radio range than LA, it is more suitable to use absolute (un-normalized) value in our case. However, for comparison purpose, we present results (error) in relative (normalized) value if necessary.

![Figure 1. Evaluation Scenario.](image)
Table 1. Comparison of Landscape and MDS-MAP(P,R) in an example.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MDS-MAP(P,R)</th>
<th>Landscape</th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes (n)</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>connectivity range error</td>
<td>27.13</td>
<td>N/A</td>
</tr>
<tr>
<td>range error</td>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>total samples</td>
<td>N/A</td>
<td>240</td>
</tr>
<tr>
<td>samples per round</td>
<td>N/A</td>
<td>15</td>
</tr>
<tr>
<td>absolute error</td>
<td>34.14</td>
<td>28.14</td>
</tr>
<tr>
<td>CPU time per node</td>
<td>9.54 sec.</td>
<td>0.27 sec.</td>
</tr>
<tr>
<td>comm. overhead</td>
<td>N/A</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Computation Overhead:** In Landscape, each sensor individually estimates its position, thus its computation complexity is $O(n)$ ($n$ is the number of the nodes); which is same as MDS-MAP(P,R) [50]. However, to get more information and make the comparison more intuitive, 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 MB DDR SDRAM) laptop with Matlab 7.0 installed. Simulations are both conducted with Calamari simulation toolkit [56], and execution time is averaged over each node in seconds.

- **Communication Overhead:** Communications are generally more energy-consuming than computations. One major advantage of Landscape is that it introduces zero sensor-to-sensor communication overhead. Our current simulation did not calculate how much communications were involved in MDS-MAP(P,R), however, obviously, the sensor-to-sensor communication overhead is at least $O(n)$.

For the scenario we investigated here, sensor nodes are assumed to be deployed on a 2-D plane. Thus in our simulations, the state vector is two dimensional: $\{x_1(n), x_2(n)\}$. Since this is a static system, it is reasonable to set the process noise variance matrix with small values: $Q = diag(10^{-6}, 10^{-6})$. And the variance matrix for measurement $R$ is simply $((\text{range.error})^2)$.

In the following subsections, we vary different parameters, such as range error, density, number of beacon samples, network size, etc., to investigate how Landscape performs in terms of accuracy and computation cost.

5.3 Results vs. Range Error and Density

In this subsection, we investigate the performance of Landscape against various range errors. Landscape does not depend on the sensor density/connectivity. However, to compare it with MDS-MAP(P,R), we experimented with different density values by adjusting the radio range $(R)$ of the sensor nodes, since the accuracy of MDS-MAP(P,R) highly relies on sensor-to-sensor connectivity. For the simulations with Landscape, we let LA broadcast 15 beacons per round and totally send out 240 beacons (in 16 rounds). We have used 10 anchor nodes for all the simulations of MDS-MAP(P,R). The results of this group of experiments are shown as Figure 2, 3 and 4. The accuracy of Landscape and MDS-MAP(P,R) is compared in Figure 2 with normalized position error, and in Figure 3 with un-normalized position error. Figure 4 shows the average CPU time per node used by the two algorithms. In all three figures, charts labeled as (a) are for range error of 5% and 10%, while the charts labeled as (b) are for range error of 15% and 20%. As shown in these figures, Landscape outperforms MDS-MAP in accuracy for all the cases, while using much less CPU time than MDS-MAP. Landscape keeps a constant value of 0.27 second CPU time for each node, while for MDS-MAP(P,R), the averaged CPU time per node grows very fast when density is increased to reduce estimation errors. Table 1 compares MDS-MAP(P,R) with Landscape by listing the parameters and results for an example case.

5.4 Results vs. Number of Beacon Samples

In this subsection, we investigate the effect of LA strategy influence on the performance of Landscape. As expected, the accuracy of Landscape increases with the number of beacon samples it used for position estimation, since the

---

2For example, reference [40] shows the energy consumed in transmitting a 1 kilobit packet over 100m is approximately the same as performing 3M CPU instructions on prototype wireless sensor nodes.
number of iterations of the UKF-based positioning algorithm is equal to the number of samples. However, more samples not only increase cost of the LA, but also introduces more computation burden for each sensor node. On the other hand, if a node has only enough power to support a specific number of computation iterations (in UKF-based algorithm), we want to know which strategy is better for it, small number of $\text{samples per round}$ with more rounds, or large number of $\text{samples per round}$ with less rounds? In this group of experiments, we varied the value of $\text{samples per round}$ and $\text{total samples}$ both from 15 to 480. As shown in Figure 5, with a fixed $\text{samples per round}$ value, positioning error decreases if we use more samples. However, note that, after 120, the accuracy increases slowly with the $\text{total samples}$; with a fixed $\text{total samples}$ value, however, a smaller value of $\text{samples per round}$ yields slightly higher accuracy than a larger one. The computation cost increases linearly with $\text{total samples}$, almost independent of the value of $\text{samples per round}$, which is demonstrated in Figure 6. The simulation gives us helpful information from two aspects: (1) sensor node prefers the LA to send a moderate number of samples per round; (2) sensor nodes can individually make tradeoffs between accuracy and computation cost by using partial or all samples.
5.5 Results vs. Network Size

We have conducted simulations on different network sizes to investigate the scalability of Landscape. Figure 7 shows the CPU time per node used by Landscape and MDS-MAP(P,R) for test cases with different number of
nodes. From Figure 4, we know that the CPU time per node of MDS-MAP(P,R) increases rapidly with connectivity. In this experiment, we have kept a roughly constant connectivity (19.5 20.5) when we generates test cases. The range error was set as 10%. As shown in Figure 7, Landscape uses a constant CPU time for each node despite of the number of nodes and range error; while MDS-MAP(P,R) introduces more computation cost on each node when network size increases or measurements have larger noise. Communication cost is at least as important as, if not more than, computation cost when scalability is evaluated. Landscape does not rely on interactive communications among sensors. Thus it is fully scalable in terms of this performance metric.

5.6 Results vs. Irregularity

Since the working of Landscape does not rely on neighborhood, it is insensitive to node density and network topology. In this subsection, we use some simple cases to demonstrate the robustness of Landscape to two kinds of irregularity: (1) non-uniform density; (2) irregular shape. As in previous sections, we still use MDS-MAP(P,R) as the reference. MDS-MAP(P,R) is a distributed algorithm, so it works well against moderate irregular situations. However, as a fundamental requirement to neighborhood measurement based algorithms, it is necessary to keep certain level of connectivity everywhere to make them work well. Figure 8 shows the results of MDS-MAP(P,R) and Landscape for a case with non-uniform node density. In the figures, small circles represent the original location of sensor nodes, while small arrows point to the estimated positions. In this case, the nodes which are closer to the sensor field center have higher density than those which are further from the center. Although the average connectivity is as high as 26.8, estimated position of some nodes which are at the fringe of the sensor field has been drifted away with significant errors. Figure 9 presents the results for a case of irregular shape, in which sensor nodes were deployed in a triangular area with a node density of 12.0. MDS-MAP(P,R) worked well for most sensor nodes in this case, while some nodes at the corner got large errors since they have less connectivity than others. Landscape performed well for both cases, and in fact, it can work well against cases with more irregularity.

5.7 Discussion

Besides the advantages of Landscape we demonstrated in above subsections, we have two observations to state:

- At current stage, Landscape does not work for the situations where only proximity is available (MDS-MAP(P,R) can work with these situations). However, this functionality can be extended in Landscape if the LA exploits more complex moving trajectories than the scenario we described here. This is one part of our future work.

- As an alternative, if more complex moving trajectories are employed so that the LA may move through a sensor node’s vicinity within a short distance, TDoA measurements may be used to achieve higher accuracy in range measurement. However, this approach will incur extra equipment cost.

6 Conclusions and Future Research Directions

This paper considers the sensor localization problem as a functional dual of target tracking. From this new perspective, we construct an on-line estimation method called Landscape for sensor localization with a localization assistant. The state information is obtained using the Unscented Kalman Filter. Our Landscape method has the following clear advantages: (1) It needs no inter-sensor and sensor-to-LA communications. It listens passively to the localization assistant. Because there is no communication overhead for localization purposes, little power and communication resources is consumed. (2) It is fully distributed, allowing high scalability to large networks. (3) It has high accuracy, and the algorithm is easy to implement with low computational complexity and low cost. (4) It is resistant to range errors, and is suitable for sensor networks with arbitrary density and topology. By experimentally comparing to the MDS method, Landscape demonstrated superior performance.
In our future research, we will extend the localization of static sensor network to that of slow-moving sensor network (compared to the speed of the localization assistant). We will investigate the situations where more complex LA moving trajectories can benefit. The performance of Landscape will be studied theoretically. We shall also investigate localization-based protocols for enhanced resilience and data rate.

Acknowledgement

We thank Drs. Y. Zhang and M. Fromherz at PARC, and Dr. V. Raghavan at DARPA for providing us the MDS-MAP codes for comparisons. We also thank Dr. Y. Shang at the Univ. of Missouri at Columbia for providing us information on the MDS-MAP codes.

References


[34] R. Nagpal, "Organizing a Global Coordinate System from Local Information on an Amorphous Computer”, A.I. Memo 1666, MIT A.I. Laboratory, August 1999.
