Landscape: A High Performance Distributed Positioning Scheme for Outdoor Sensor Networks

Liqiang Zhang, Qiang Cheng,Yingge Wang, and Sherali Zeadally

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.**

Index Terms— **Wireless Sensor Networks, Localization algorithm, Unscented Kalman Filter.**

I. INTRODUCTION

Future wireless sensor networks (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. 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 reduced. Location-aware sensors may also help to highly enhance the efficiency of routing protocols [17], [28] by reducing costly message flooding. However, installing a global positioning system (GPS) receiver on 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.

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 (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 [2], [6], [12], [24] or even anchor-free [10], [22]. 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 [8], [22], [25], [30], or more generally, a constrained optimization problem [6], [24]. 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 [6] may involve high computation complexity becoming usable only when the application permits deployment of a central processor to perform location estimation [20]; on the other hand, some distributed solutions (such as AFL [22], 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* [5].

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 one or more 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) [15] 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 [19]. 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 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 [26], [27].
- *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 [7], [12], [16]. 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*. 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 [26], [27] 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.

II. Related Work

Sensor localization has attracted significant research effort in recent years, and various approaches have been proposed [2], [6], [4], [10], [12], [22], [24]. 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 [24], [27] or global information [26]. There are different ways to categorize the existing approaches by the techniques used in these phases. We made a thorough survey of existing methods by classifying the existing methods according to the raging techniques (ToA/TDoA, AoA, and RSS) in our technical report (online) [31]. In this paper, however, due to the space limitation, we only present a brief review of some most related work. A short discription on MDS-MAP related algorithms [26], [27] is given in the following subsection, then we take a glance at Bayesian techniques for robot location estimations.

A. Multidimensional Scaling for Localization

Multidimensional scaling (MDS) [1] has recently been successfully used to resolve sensor localization problem [26], [27], [13]. MDS can be seen as a set of data analysis techniques that display the structure of distance-like data as a geometrical picture [1]. 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 [13]. Shang et al. first proposed MDS-MAP to use MDS in sensor location problem in [26]. 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) [27] 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 [27]. 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%.

B. Bayesian Techniques for Robot Localization

Bayesian techniques have been widely investigated in the context of robot localization [9], [11]. Recently, grid based Markov localization [3], particle filter (a.k.a. sequential Monte Carlo) [7], real-time particle filters [16] 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) [12]. 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 [14] in location estimation, which is easy to implement and has high accuracy in the presence of nonlinear observation functions and non-Gaussian 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.

III. Landscape Localization Methodology and Model

A. 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 Landscape 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.

B. Landscape Localization Model

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

$$\mathbf{x}(n) = \{x_1(n), x_2(n), x_3(n)\}. \tag{1}$$

And we have the following dynamic state and observation equations:

$$\begin{aligned} \mathbf{x}(n) &= \mathbf{f}(\mathbf{x}(n-1)) + \mathbf{w}(n), \\ \mathbf{y}(n) &= \mathbf{g}(\mathbf{x}(n)) + \mathbf{v}(n), \end{aligned} \tag{2}$$

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 observation noise sequences.

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

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

with $\mathbf{w}_i(n)$ modeling the small position perturbation due to the wind or other environmental effects. Our algorithm 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:

$$y(n) = \sqrt{(\Delta x(n))^2 + (\Delta y(n))^2 + (\Delta z(n))^2} + v(n). \tag{4}$$

Here $\Delta x(n) = x^t(n) - \mathbf{x}_1(n)$, $\Delta y(n) = y^t(n) - \mathbf{x}_2(n)$, $\Delta z(n) = z^t(n) - \mathbf{x}_3(n)$. $(x^t(n), y^t(n), z^t(n))$ 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,

$$[x^t(n), y^t(n), z^t(n)]^T = \mathbf{h}(n\Delta T; x, y, z), \tag{5}$$

where ΔT is the time interval of broadcasting, and $\mathbf{h}(\cdot)$ 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 $c_z - \mathbf{x}_3(n) \triangleq \Delta h$. 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 filter (UKF) which is an elegant and computationally efficient recursive state estimation technique and well suited to our applications.

IV. Landscape State Estimation Via Unscented Kalman Filter

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 section, 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 $\mathbf{g}(\cdot)$, the unscented transformation (UT) [14], [18] 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 $\mathbf{g}(\cdot)$ for Gaussian noise sequences, or the second-order for arbitrary noise distributions. Higher-order approximation can also be captured with extended algorithms [15]. At the same time, UT uses the same order of calculations as linearization. We shall apply the unscented Kalman filter (UKF) [14] in our Landscape system.

A. Unscented Transformation

The UT has been developed to handle low-order statistics of random variables that undergo a nonlinear transform $g(\mathbf{x})$. The knowledge of higher order information can also be partially incorporated into the *sigma point* set. Let D_x be the dimension of $\mathbf{x}$, μ_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 $\mathcal{S} = \{\mathbf{x}_k, W_k : k = 0, \cdots, 2D_x\}$, with W_0 denoting the weight on the mean point:

$$\begin{aligned} \mathbf{x}_0 &= \mu_x, \quad W_0 = W_0, \\ \mathbf{x}_k &= \mu_x + \left(\sqrt{\frac{D_x}{1-W_0}K_x}\right)_k, \quad W_k = \frac{1-W_0}{2D_x}, \\ \mathbf{x}_{k+D_x} &= \mu_x - \left(\sqrt{\frac{D_x}{1-W_0}K_x}\right)_k, W_{k+D_x} = \frac{1-W_0}{2D_x} \end{aligned} \tag{6}$$

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

2) Propagate the sigma points through the nonlinear transformation

$$\mathbf{y}_k = g(\mathbf{x}_k), \quad k = 0, \cdots, 2D_x. \tag{7}$$

3) Calculate the mean and variance of the transformed points

$$\begin{aligned} \mu_y &= \sum_{k=0}^{2D_x} W_k \mathbf{y}_k, \\ K_y &= \sum_{k=0}^{2D_x} W_k (\mathbf{y}_k - \mu_y)(\mathbf{y}_k - \mu_y)^T. \end{aligned} \tag{8}$$

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

B. 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 $\mathbf{x}^a(n) = [\mathbf{x}^T(n), \mathbf{w}^T(n), \mathbf{v}^T(n)]^T$. The resultant augmented vector is of dimension $D_a = D_x + D_w + D_v$. The process and observation models for the augmented vector $\mathbf{x}^a(n)$ from (2) are

$$\begin{aligned} \mathbf{x}^a(n) &= \mathbf{f}^a(\mathbf{x}^a(n-1)), \\ \mathbf{y}^a(n) &= \mathbf{g}^a(\mathbf{x}^a(n)). \end{aligned} \tag{9}$$

The UKF is implemented as follows [14]:

1) Initialization:

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

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

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

$$\mathbf{x}_k^a(n) = \mathbf{f}^a(\mathbf{x}^a(n-1)), \tag{11}$$

$$\mu_x^a(n) = \sum_{k=0}^{2D_a} W_k \mathbf{x}^a(n), \tag{12}$$

$$K^a(n) = \sum_{k=0}^{2D_a} W_k (\mathbf{x}^a(n) - \mu^a(n)) \times (\mathbf{x}^a(n) - \mu^a(n))^T, \tag{13}$$

$$\mathbf{y}(n) = \mathbf{g}^a(\mathbf{x}^a(n)), \tag{14}$$

$$\mu_y(n) = \sum_{k=0}^{2D_a} W_k \mathbf{y}(n). \tag{15}$$

c) Update:

$$P_y^a(n) = \sum_{k=0}^{2D_a} W_k(\mathbf{y}(n) - \mu_y(n)) \times (\mathbf{y}(n) - \mu_y(n))^T, \quad (16)$$

$$P_{xy}^a(n) = \sum_{k=0}^{2D_a} W_k(\mathbf{x}^a(n) - \mu_x^a(n)) \times (\mathbf{y}(n) - \mu_y(n))^T, \quad (17)$$

$$W(n) = P_{xy}^a(n)(P_y^a(n))^{-1}, \quad (18)$$

$$\mu_x(n) = \mu_x(n-1) + W(n)(\mathbf{y}(n) - \mu_y(n)), \quad (19)$$

$$K_x(n) = K^a(n) - W(n)P_y^a(n)W^T(n). \quad (20)$$

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 [14] [18]. 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.

V. Performance Evaluation

A. 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:

$$x^t(n) = c_x + R_{LA}\cos(2\pi/samples\_per\_round * (n-1)),$$
$$y^t(n) = c_y + R_{LA}\sin(2\pi/samples\_per\_round * (n-1)), \quad (21)$$
$$z^t(n) = c_z.$$

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. 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

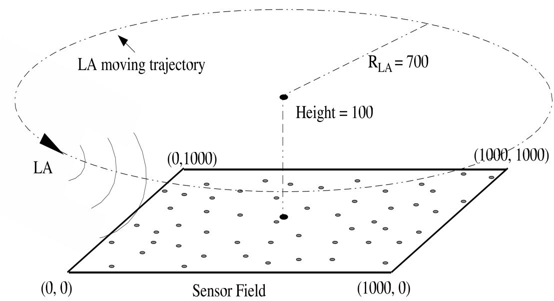


Figure 1. Evaluation Scenario.

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 [26], [27]. A random noise is added to the true distance as following:

$$\hat{d} = d * (1 + randn(1) * range\_error) \quad (22)$$

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

B. Evaluation Metrics and Parameters

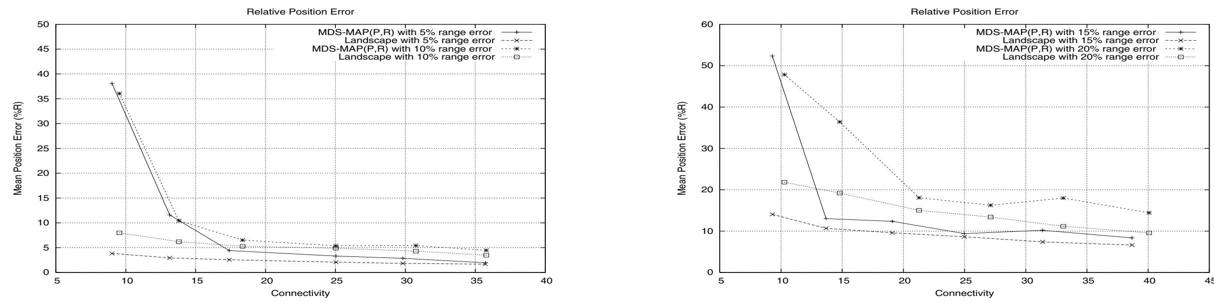
We code Landscape in Matlab for simulation purposes. To make our proposal comparable to other positioning schemes, we have interfaced our algorithm to the localization simulation toolkit designed as part of Berkeley's Calamari project [29]. We selected MDS-MAP as a reference for performance evaluation, since to the best of our knowledge, MDS-MAP is one of the algorithms which give 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.
- **Computation Overhead:** In Landscape, each sensor individually estimates its position, thus its computation complexity is $\bigcirc(n)$ (n is the number of the nodes); which is same as MDS-MAP(P) [27]. 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 [29], and execution time is averaged over each node in seconds.

TABLE I

COMPARISON OF LANDSCAPE AND MDS-MAP(P,R) IN AN EXAMPLE.

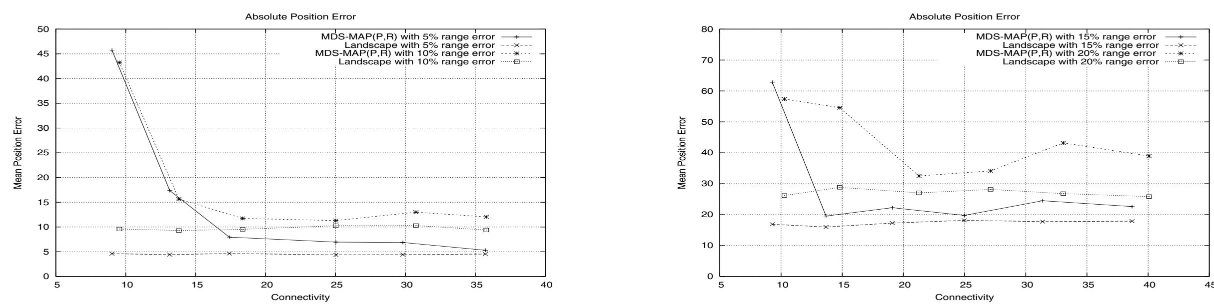
	Parameters					Results		
	# of nodes (n)	connectivity	range error	total samples	samples per round	absolute error	CPU time per node	inter-sensor communication overhead
MDS-MAP	200	27.13	20%	N/A	N/A	34.14	9.41 sec.	$\bigcirc(n)$
Landscape	200	N/A	20%	240	15	28.14	0.27 sec.	0



(a) **(b)**

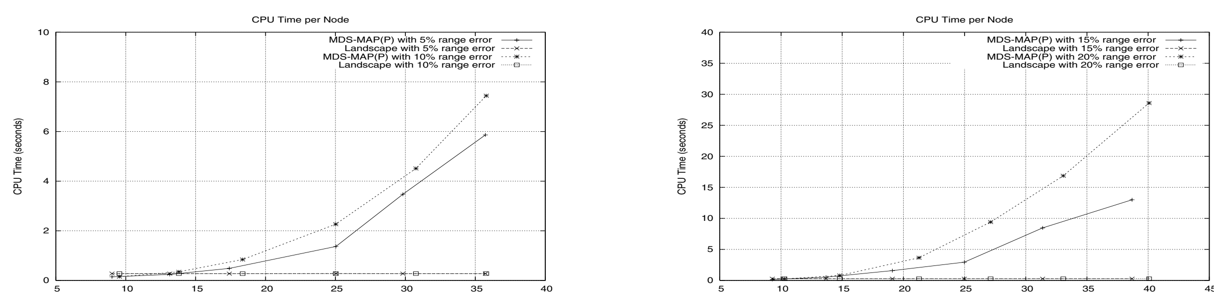
Figure 2. Relative Position Error vs. Range Error and Density.

Landscape yields better accuracy than MDS-MAP(P,R) for different range errors. MDS-MAP(P,R) highly relies on connectivity, while Landscape has almost constant accuracy with variant connectivity. The curves for Landscape are not horizontal lines simply because results are normalized to the sensor radio range R, which is adjusted to have variant connectivity.



(a) **(b)**

Figure 3. Absolute Position Error vs. Range Error and Density.



(a) **(b)**

Figure 4. CPU Time vs. Range Error and Density.

- **Communication Overhead:** Communications are generally more energy-consuming than computations [21]. One major advantage of Landscape is that it introduces zero sensor-to-sensor communication overhead. Our current simulations did not calculate the communications involved in MDS-MAP, however, obviously, it is at least $\bigcirc(n)$.

Since MDS-MAP(P,R) generally achieves better accuracy than MDS-MAP(P) at the cost of higher computation overhead, MDS-MAP(P,R) is chosen as the reference when we evaluate the accuracy, while MDS-MAP(P) is chosen when we investigate the computation overhead. 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 $((range\_error)^2)$.

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.

C. 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, we experimented with different density values by adjusting the radio range (R) of the sensor nodes, since the accuracy of MDS-MAP 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. The results of this group of experiments are shown as Figure 2, 3 and 4. The accuracy of Landscape and MDS-MAP 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, 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.

D. Results vs. Network Size

We have conducted simulations on different network sizes to investigate the scalability of Landscape. Figure 5 shows the CPU time per node used by Landscape and MDS-MAP for test cases with different number of nodes. From Figure 4, we know that the CPU time per node of MDS-MAP 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 5, Landscape uses a constant CPU time for each node despite of the number of nodes and range error; while MDS-MAP 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.

E. 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 based algorithms, it is necessary to keep certain level of connectivity everywhere to make them work well. Figure 6 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 7 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.

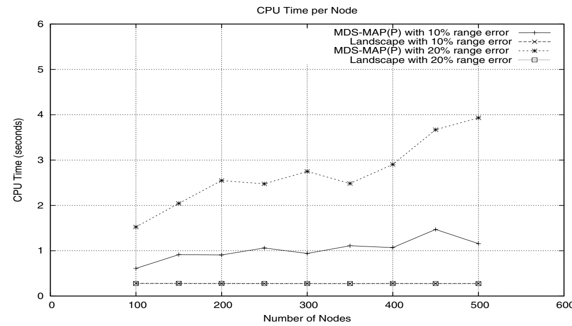
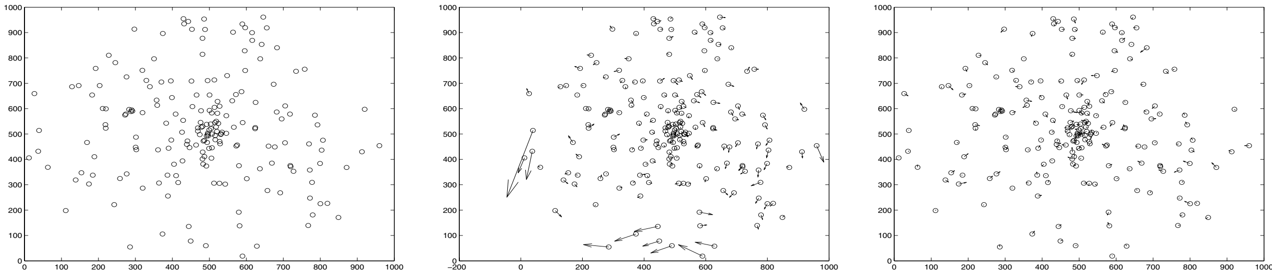


Figure 5. CPU Time per Node vs. Network Size.

VI. Conclusions and Future Research Directions

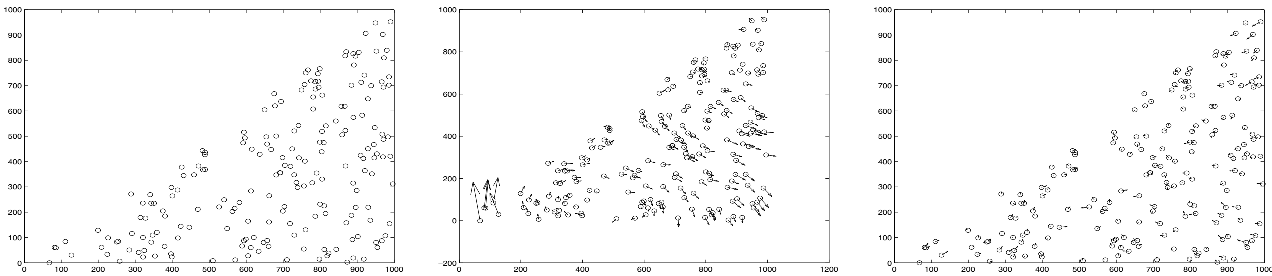
This paper considers the sensor localization problem as a functional dual of target tracking. From this new perspective, we construct an on-line estimation method called Landscape for sensor localization with a localization assistant. The state information is obtained using the Unscented Kalman Filter. 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 trajectory can benefit. The performance of Landscape will be studied theoretically. We shall also investigate localization-based protocols for enhanced resilience and data rate.



Original Map **Result of MDS-MAP(P,R)** **Result of Landscape**

Figure 6. A Case of Non-uniform Node Density.



Original Map **Result of MDS-MAP(P,R)** **Result of Landscape**

Figure 7. A Case of Irregular Shape.

REFERENCES

[1] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Applications*, Apringer-Verlag, New York, 1997.

[2] N. Bulusu, J. Heidemann and D. Estrin, "Density Adaptive Algorithms for Beacon Placement in Wireless Sensor Networks," *in Proceedings of IEEE ICDCS'01*, Phoenix, AZ, April 2001.

[3] W. Burgard, A. Derr, D. Fox, and A. Cremers, "Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach. *in Proceedings of IEEE/RSI IROS'98*, Victoria, Canada, October 1998.

[4] X. Cheng, A. Thaeler, G. Xue and D. Chen, "TPS, A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks," *in Proceedings of IEEE INFOCOM'04, Hong Kong*, China, March 2004.

[5] K. Chintalapudi, A. Dhariwal, R. Govindan and G. Sukhatme, "Ad-Hoc Localization Using Ranging and Sectoring," *in Proceedings of IEEE INFOCOM'04*, Hong Kong, China, March 2004.

[6] L. Doherty, K. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," *in Proceedings of IEEE INFOCOM'01*, Anchorage, Alaska, April 2001.

[7] F. Dellaert, D. Fox, W. Burgard and S. Thrun, "Monte Carlo Localization for Mobile Robots," *in Proceedings of ICRA'99*,Detroit, Michigan. May 1999.

[8] T. Eren, D. Goldenberg, W. Whilteley, Y. Yang, A. Morse, B. Anderson and P. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization," *in Proceedings of IEEE INFOCOM'04*, Hong Kong, China, March 2004.

[9] D. Fox, J. Hightower, L Liao, and D. Schulz, "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing*, Vol. 2 No. 3, pp. 24-33, 2003.

[10] T. He, C. Huang, B. Blum, J. A. Stankovic and T. F. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *in Proceedings of ACM MOBICOM'03*, San Diego, CA, September 2003

[11] A. Howard, M. Mataric and G. Sukhatme, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation," *in Proceedings of IROS'02*, EPFL, Switzerland, October 2002.

[12] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," *to appear in ACM MOBICOM'04*, Philadelphia, Pennsylvania, September, 2004.

[13] X. Ji, H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling," *in Proceedings of IEEE INFOCOM'04*, Hong Kong, China, March 2004.

[14] S. Julier, "A skewed approach to filtering" *in Proc. AeroSense: Int. Symp. Aerospace/Defense Sensing, Simulation, and Controls*, vol. 3373, pp. 56-65, Apr. 1998.

[15] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004.

[16] C. Kwok, D. Fox, M. Meila, "Adaptive Real-time Particle Filters for Robot Localization," *in Proceedings of ICRA'03*, Taipei, Taiwan, September 2003.

[17] J. Li, J. Jannotti, D.S.J. DeCouto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *in Proceedings of ACM MOBICOM'00*, Boston, MA, August, 2000.

[18] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter". *Technical Report CUED/F-INFENG/TR 380*, Cambridge University Engineering Department, Aug. 2000.

[19] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," *in Proceedings of IEEE INFOCOM'03*, San Francisco, CA, April 2003

[20] N. Patwari, A. Hero III, M. Perkins, N. Correal, and R. O'Dea. "Relative location estimation in wireless sensor networks". *IEEE Trans. on Sig. Proc., Special Issue on Sig. Proc. in Networking*, 51(8):2137-2148, August 2003.

[21] G. Pottie and W. Kaiser, "Embedding the internet: wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, May 2000.

[22] N.B. Priyantha, H. Balakrishnan, E. Demaine and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks," *LCS Technical Report #892*, MIT, April 2003.

[23] A. Savvides, C.-C. Han and M.B. Srivastava, "Dynamic Fine- Grained Localization in Ad-Hoc Networks of Sensors," *in Proceedings of ACM MOBICOM'01*, Rome Italy, July 2001.

[24] A. Savvides, H. Park, and M. Srivastava, "The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems," *in Proceedings of ACM WSNA02*, Atlanta, GA, September 2002.

[25] J. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," *in Proceedings of 17th Allerton Conference on Communications, Controls, and Computing (1979)*, pp. 480-489.

[26] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization From Mere Connectivity". *in Proceedings of ACM MOBIHOC'03*, Annapolis, MD, June 2003.

[27] Y. Shang, W. Ruml, "Improved MDS-Based Localization," *in Proceedings of IEEE INFOCOM'04*, Hong Kong, China, March 2004.

[28] I. Stojmenovic, "Location updates for efficient routing in ad hoc wireless networks," *Handbook of Wireless Networks and Mobile Computing*, Wiley, 451-471, 2002.

[29] K. Whitehouse, "Calamari: a sensor field localization system," *http://www.cs.berkeley.edu/ kamin/calamari.*

[30] Y. Yemini, "Some Theoretical Aspects of Position-location Problems," *in Proceedings of 20th Annual IEEE Symposium on Foundations of Computer Science (1979)*, pp. 1-8.

[31] L. Zhang, Q. Cheng, Y. Wang and S. Zeadally, "A Novel High Performance Distributed Sensor Positioning System," online technical report, *http://www.ece.eng.wayne.edu/∼qcheng/Landscape.pdf.*