

# Landscape-3D: A Robust Localization Scheme for Sensor Networks over Complex 3D Terrains

Liqiang Zhang

Dept. of Computer & Information Sciences  
Indiana University South Bend,  
South Bend, IN 46634, USA

Xiaobo Zhou

Dept. of Computer Science  
University of Colorado at Colorado Springs,  
Colorado Springs, CO 80918, USA

Qiang Cheng

Siemens Medical Solutions  
Princeton, NJ 08540, USA

**Abstract**—Despite the fact that sensor networks could often be deployed over three-dimensional (3D) terrains, most approaches on sensor localizations are designed and evaluated considering only two-dimensional (2D) applications. On the other hand, being the foundation of the most previous localization solutions, reliable and sufficient neighborhood-measurements are often hard to achieve for sensor nodes deployed in complex 3D terrains, which makes it difficult to extend those solutions into 3D applications. In the paper, we introduce a robust 3D localization solution called Landscape-3D, in which we treat the localization problem from a novel perspective by taking it as a functional dual of target tracking. Besides several nice features, such as high scalability, high accuracy, zero sensor-to-sensor communication overhead, low computation overhead, etc., one of the most important advantages of Landscape-3D is that it works totally independent of node densities and network topologies, which makes it robust to complex 3D environments. Our simulation model involves various 3D scenarios. Experimental results demonstrate that Landscape-3D is a robust localization approach for sensor networks deployed in complex 3D terrains.

## I. INTRODUCTION

It is believed that wireless sensor networks (WSNs) will extend our sensory capability to every corner of the world. 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. For most of these applications, it is important for the sensor nodes to be aware of their own locations. Sensed data are often more meaningful when they are associated with spatial coordinates. Location-aware sensors may also help to highly enhance the efficiency of routing protocols [16], [22] 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 size, the battery, and the cost constraints of sensor nodes.

As a fundamental problem in sensor networks, the self-localization of sensor nodes has recently attracted massive attentions from both academia and industry. The constraints of sensor nodes, such as limited computation power, limited battery capacity, requirements of small size and low cost, have made the sensor's location discovery a very challenging research issue. A good solution for this problem has to be

distributed, light-weight, energy-efficient, and low-cost [4].

Despite of many research proposals on sensor localization problem [1], [2], [6], [8], [12], [9], [11], [18], [19], most of them are designed and evaluated considering only 2-dimensional (2D) applications where sensor networks are deployed over flat terrains. In real applications, however, sensor networks could often be deployed over 3-dimensional (3D) terrains. For example, a surveillance network deployed in a mountainous battlefield, a sensor network floating in the air for pollution monitoring, or a structural monitoring network mounted on a bridge. These 3D applications bring more than just one extra dimension to the localization problem. In a sensor network deployed over complex 3D terrains, network topologies could be much more complex than 2D cases, which requires sensor localization schemes to be more robust to network irregularities.

Besides the fact that there is no localization result reported for 3D sensor networks so far, most current approaches are difficult to be extended to three dimensions. Among the numerous proposals for sensor localization, most of them are based on neighborhood-measurements [6], [11], [12], [20], [21], [19], in which the location of a sensor node is estimated utilizing measured distances or angles from its neighbor nodes. In neighborhood-measurement based localization methods, a sensor node is able to calculate its own position only if it has sufficient neighbors. As pointed out in [5], they begin to perform acceptably only when the node densities are *well beyond the density required for network connectivity*. In networks over complex 3D terrains, because of non-uniform node densities, irregular topologies, and obstructions, it is often too optimistic to assume every sensor node being able to achieve sufficient neighborhood-measurements.

Motivated by this observation, in this work, we introduce a robust 3D sensor localization scheme called Landscape-3D, which solves the localization problem from a different perspective than existing approaches by taking it as a functional dual of target tracking. Traditional target tracking solutions utilize one or more static location-aware sensors to track and predict the position (and/or speed) of a moving target. In Landscape-3D, by introducing a mobile location-assistant (LA, could be aircraft, balloon, robot, vehicle, etc.), we let each location-unaware sensor discover its position by passively observing the moving, location-aware LA (with the

TABLE I  
COMPARISON OF LANDSCAPE-3D AND NEIGHBORHOOD-MEASUREMENT-BASED LOCALIZATION METHODS.

	<b>Neighborhood-measurement-based localization methods</b>	<b>Landscape-3D</b>
<b>Accuracy</b>	Depends on algorithms as well as node densities, from low to high. A high accuracy is usually at the cost of high computation cost.	High.
<b>Scalability</b>	Depends on algorithms, from low (centralized) to high (distributed).	High, each node discovers its location with its own measurements and calculations.
<b>Computation overhead</b>	Depends on algorithms as well as node densities, from low to high.	Low.
<b>Communication overhead</b>	High to very high.	Low, introduces ZERO sensor-to-sensor communications.
<b>Robustness to node densities</b>	Weak, accuracy highly depends on node densities.	Strong, does not rely on node densities.
<b>Robustness to network topology</b>	Weak, not work well for irregular topologies.	Strong, works well for all topologies.
<b>Robustness to range errors</b>	Depends on algorithms, from weak to strong.	Strong.
<b>Ranging Techniques</b>	RSS or ToA (Time of Arrival) or (TDoA) or AoA (Angle of Arrival).	RSS.
<b>Implementation cost</b>	Depends on ranging techniques used, from low to high.	Low, no need of special equipment for sensor nodes, and the cost of a single LA is amortized on each sensor node.
<b>Ability to Support 3D Applications</b>	Weak, no results reported yet.	Strong.

GPS or pre-defined moving path). We resolve this functional dual problem by modeling and utilizing an Unscented Kalman Filter (UKF) [14] based algorithm. One scenario frequently mentioned in the literature is that sensor nodes are deployed by an aircraft. Landscape-3D fits well with (but not limited to) this kind of sensor applications. We can simply let the aircraft cruise several rounds above the sensor field in 3D terrains, 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-3D is a novel extension of the Landscape scheme presented in our previous work [25], [26]. By introducing a new 3D network model and three dimensional LA moving trajectories, Landscape-3D successfully extends the ability of landscape to support 3D applications while it does not involve significant changes on the core UKF-based algorithm. Thus, Landscape-3D introduces a small amount (less than 10%) extra computation overhead. It also inherits almost all the advantages of Landscape. Table I summarizes the features of Landscape-3D using neighborhood-measurement methods as the reference <sup>1</sup>. In the rest of this work, however, instead of demonstrating all these features, we focus our attention on the support of 3D sensor localizations. In our experiments, we have constructed several 3D scenarios to simulate complex terrains. The beacon loss situation has also been considered in simulations. In a stringent environment, some beacons may not be reachable to all sensor nodes; each sensor may individually observe a different incomplete beacon set depending on many factors, such as obstructions, time-varying link qualities, hostile radio-jamming attacks, etc.

<sup>1</sup>The effectiveness and advantages of Landscape were demonstrated in our previous work [25], [26] through a comprehensive evaluation study using a state-of-art neighborhood-measurement sensor localization method – MDS-MAP [20], [21] – as the reference.

We address this issue by including two circumstances in our simulation studies: randomly uniformly dropped beacons, and randomly bursty dropped beacons. Our simulations reveal that Landscape-3D is a robust localization approach for sensor networks deployed over complex terrains. It works reasonably well even if significant part of beacons are unreachable to sensor nodes. To the best of our knowledge, this is the first effort reporting localization results for 3D cases.

The rest of this paper is organized as follows: A short review of related work is presented in the next section. Section III presents the 3-dimensional network model. We describe the Landscape-3D localization system in section IV. Experimental results of our proposed system against complex terrains are presented and discussed in section V. Finally, concluding remarks are given in section VI.

## II. RELATED WORK

Sensor localization has attracted significant research efforts in recent years, and various approaches have been proposed [2], [6], [9], [11], [12], [18], [19]. 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 are based on neighborhood-measurements and 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 [19], [21] or global information [20]. There are different ways to categorize the existing approaches by the techniques used in those phases. Please refer to [26] for a rather complete survey of existing methods which classifies the existing methods according to the ranging techniques (ToA/TDoA, AoA, and RSS). In this paper, we only present a brief review of most related work. A short description on MDS-MAP related algorithms [21] 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) has recently been successfully used to resolve sensor localization problem [12], [20], [21]. MDS can be seen as a set of data analysis techniques that display the structure of distance-like data as a geometrical picture. 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. Shang et al. first proposed MDS-MAP to use MDS in sensor location problem in [20]. 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) [21] 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).

As the state-of-the-art neighborhood-measurement based approach, MDS-MAP(P,R) has demonstrated impressive performance. However, as we will demonstrate in Section V, this method is quite sensitive to node densities and network topologies, thus it is difficult to extend this method into a solution for applications deployed over complex 3D terrains.

#### B. *Bayesian Techniques for Robot Localization*

Bayesian techniques have been widely investigated in the context of robot localization [10]. Recently, grid based Markov localization [3], particle filter (a.k.a. sequential Monte Carlo) [7], real-time particle filters [15] have been proposed and shown to be successful for robot location estimation. Those 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.

Inspired by the techniques used for robot localization, Hu and Evans [11] first proposed to use sequential Monte Carlo (SMC) method for mobile sensor node localization. Their

approach is called MCL (Monte Carlo 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-3D needs only one mobile LA, and, it is mainly for static sensor networks. (2) MCL utilizes only proximity measurement, with the location estimation coarse-grained and bounded. In contrast, Landscape-3D exploits range measurement and it 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.

### III. NETWORK MODEL

We assume that the sensors are deployed randomly over a 3-dimensional monitored area. As an example, Figure 1 shows a sensor network deployed over a monitored mountain area. Each sensor node has limited resources (CPU, battery, etc), and, is equipped with an omni-directional antenna. A location assistant (LA) could be an airplane, a mobile robot, a vehicle, a balloon, etc. which is a choice up to the network designers. However, we do have the following minimal requirements on the LA:

- The LA has moving ability, being able to move around the sensor field.
- The LA is aware of its own location while it is moving, either through a GPS or a pre-defined moving path.
- The LA is able to broadcast beacons to sensor nodes; each beacon contains the LA's current location and the transmitting power used to transmit the beacon. In the rest of the paper, the term beacon's location is used to reference the location of the LA when a beacon is broadcasted.

The LA is free to leave after beacons are broadcasted. During the process, each sensor passively listens to the beacons, estimating the distance between itself and the beacon based on the measured RSS of the beacons. The localization process introduces no sensor-to-sensor communication overhead. The communication ability of sensor nodes to the LA is not assumed.

### IV. LANDSCAPE-3D LOCALIZATION METHODOLOGY, MODEL AND ALGORITHM

#### A. *Landscape-3D Localization Model*

The key idea of Landscape-3D 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 angle of arrives (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, the Landscape-3D system exploits varying positions of the LA and the corresponding sensor-to-LA distances to dynamically determine the positions of sensor nodes.

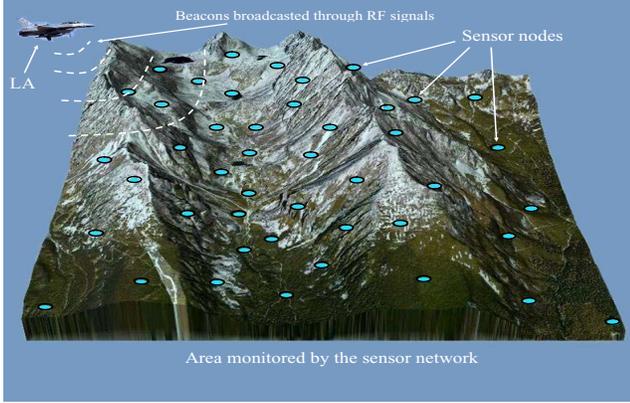


Fig. 1. An example sensor network deployed over 3D terrains.

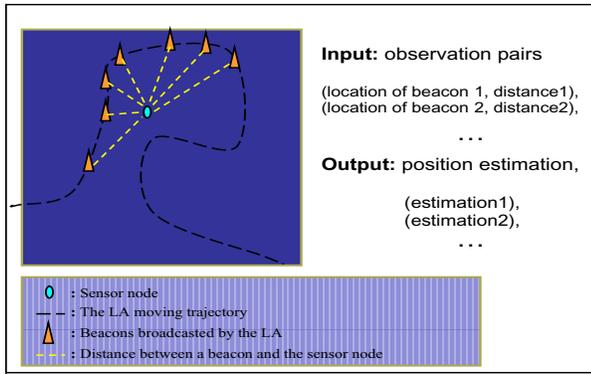


Fig. 2. Landscape-3D methodology illustrated in 2-dimensions.

The sensor position is determined by solving the associated state evolution and observation dynamics of the positions of the LA and the measured distances. Figure 2 illustrates the Landscape-3D localization methodology in a 2D manner.

For the localization problem described above, we define the state variable as the 3D position of a specific sensor node. The state of the  $i$ th sensor node at the  $n$ th iteration is:

$$\mathbf{x}_i(n) = \{x_{i1}(n), x_{i2}(n), x_{i3}(n)\}. \quad (1)$$

And we have the following dynamic state and observation equations:

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

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

Here let us consider the static sensor localization, where 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:

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

where  $\mathbf{w}_i(n)$  models the small position perturbation due to the wind or other environmental effects. Note that 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, and 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 sensors. The following observation model is used:

$$\mathbf{y}_i(n) = \sqrt{(\Delta x_{i1}(n))^2 + (\Delta x_{i2}(n))^2 + (\Delta x_{i3}(n))^2} + \mathbf{v}_i(n). \quad (4)$$

Here  $\Delta x_{i1}(n) = x_1^b(n) - x_{i1}(n)$ ,  $\Delta x_{i2}(n) = x_2^b(n) - x_{i2}(n)$ ,  $\Delta x_{i3}(n) = x_3^b(n) - x_{i3}(n)$ ; and  $(x_1^b(n), x_2^b(n), x_3^b(n))$  is the current 3D position of the LA, measured using GPS or controlled by the pre-defined path.  $\mathbf{v}_i(n)$  models the observation error, which usually comes from the RF distance estimations or the perturbations to the LA positions. We assume  $w_i(n)$  and  $v_i(n)$  are zero-mean uncorrelated noise processes.

### B. Dynamic State Estimation Via Unscented Kalman Filter

The Landscape-3D localization scheme aims at improving the sensor localization by iteratively updating the position estimates with the current observations. For the system model defined in the previous section, on-line state estimation has to be performed. Kalman filters and their variants have been designed for this purpose, but their actual performance depends heavily on the evolution 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 localization model. 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) [13], [17] 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 [14]. At the same time, UT uses the same order of calculations as linearization. The above analysis has driven us to utilize unscented Kalman filter (UKF) [13] in our Landscape-3D scheme.

The basic idea of UT is to represent the state distribution by a minimal set of carefully chosen sample points (sigma points). The UKF embeds the UT into the Kalman Filter's recursive prediction and update structure. The detailed theoretical background of UT and UKF can be found in our online technical report [26]. We skip the details here, however, due to the space limitation.

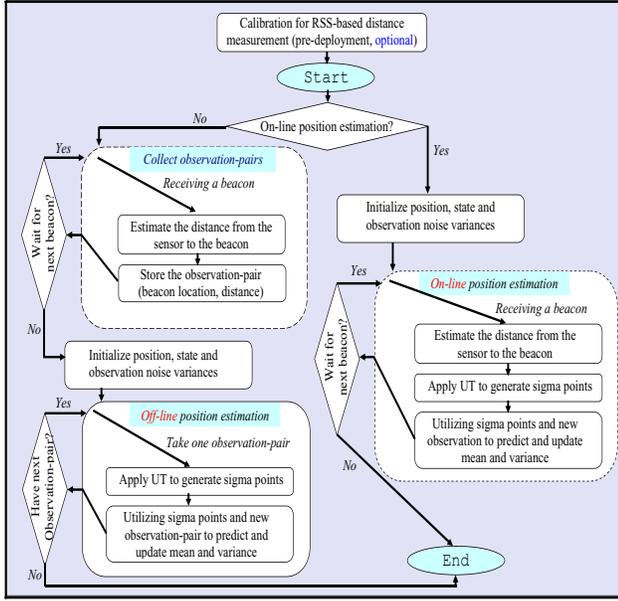


Fig. 3. Landscape-3D's UKF-based localization algorithm.

### C. UKF-based Algorithm

Figure 3 outlines the UKF-based localization algorithm. As shown in the picture, the algorithm has an optional calibration procedure, which could be done before sensor nodes are deployed. The purpose of the calibration is to improve the accuracy of RSS-based distance measurements [23]. The core of the algorithm is the iteration of state prediction and updating, which could be done either on-line or off-line. When the iterations are done off-line, each sensor node first collects all the observation-pairs (each of which contains a beacon's position and its distance from the sensor node), then executes the UKF loops to update its location estimation exploiting the constraints increasing added by each observation-pair. The off-line version of the algorithm does not have a time constraint<sup>2</sup> on each iteration of state prediction/updating, thus it is more suitable for sensor nodes that have lower computation power. However, Each sensor node needs to have several kilo-bytes memory<sup>3</sup> to temporarily store the observation-pairs, which is a reasonable requirement for most sensor applications.

Since each sensor individually calculates its own location, the computation complexity of Landscape-3D is independent of the network size. In another words, the computation overhead is  $\mathcal{O}(n)$  ( $n$  is the number of sensors in the network) in terms of the whole network or  $\mathcal{O}(1)$  in terms of each sensor. Unlike neighborhood-measurement based location methods,

<sup>2</sup>The on-line version requires the iteration for one beacon to be finished before the next beacon comes.

<sup>3</sup>As shown in the later section, for the example scenarios, 240 beacons are enough for Landscape-3D to work well. If we use 6 bytes to represent a beacon's location (3D), 2 bytes to represent the distance from the beacon to a sensor node, the observation-pair for each beacon will consume 8 memory bytes to store. for 240 beacons, we need totally 1920 bytes.

where sensors usually communicate with each other massively (for ranging measurements, and for exchanging location estimations to refine the results), Landscape-3D introduces zero sensor-to-sensor communications. The communication from sensor nodes to the LA is not needed as well. Considering communications usually are more energy-consuming than computations, this is an important advantage of Landscape-3D.

## V. EXPERIMENTAL RESULTS

In our experiments, we run Landscape-3D on various 3D terrain models in Matlab. Since previous approaches only report results for 2-dimensional localizations, it is hard to compare the performance of Landscape-3D with them for 3D cases. As a way around, besides the complex 3D terrain models, we also include some results for flat (2D) terrain models that have irregular topologies. For the 2D terrain models, we compare the performance of Landscape-3D with the most well-known neighborhood-measurement based approach – MDS-MAP. For comparison purpose, both algorithms are interfaced to the localization simulation toolkit designed as a part of the Berkeley's Calamari project [24].

Table II summarizes the metrics and parameters used in our simulations. We assume distance measurements have Gaussian noise [20], [21]. A random noise is added to the true distance as following:

$$\hat{d} = d * (1 + randn(1) * range\_error), \quad (5)$$

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

### A. Flat Terrains with Irregular Topologies

In this experiment, we use MDS-MAP as the reference to evaluate the performance of Landscape-3D for a sensor network deployed over a flat terrain. We use a square sensor field (1000 by 1000) with (0,0), (0, 1000), (1000, 1000), and (1000, 0) as its four corners. To construct an irregular network topology, we assume there is a lake in the middle of the sensor field. The lake is in round shape and has a radius of 400 with (500, 500) as its center. 200 sensor nodes are randomly deployed over the sensor field around the lake. We let an airplane be the LA. For this scenario, the LA hovers over the sensor field on a 2D plane parallel to the sensor field. The height of the airplane is a constant value, for which we used 100 feet. The LA periodically broadcasts beacon samples to sensor nodes. In this scenario, the location of the LA at time step  $n$  ( $n \geq 1$ ) is simply:

$$\begin{aligned} x_1^b(n) &= c_1 + R_{LA} \cos(2\pi/samples\_per\_round * (n - 1)), \\ x_2^b(n) &= c_2 + R_{LA} \sin(2\pi/samples\_per\_round * (n - 1)), \\ x_3^b(n) &= c_3, \end{aligned} \quad (6)$$

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

TABLE II  
METRICS AND PARAMETERS USED IN EVALUATIONS.

	Metrics and Parameters	Definition/Explanation
<b>Metrics</b>	Accuracy (position error)	The accuracy of sensor positioning is presented as the average distance between estimated positions to the true positions.
	Computation Overhead	For comparison purpose, we report the CPU time consumed (per sensor node) by position algorithms in our simulations. All simulations are conducted on a DELL precision 670 workstation (Intel Xeon 3.0GHz CPU, 2 GB DDR-2 memory).
<b>Parameters</b>	Range Error	The error introduced in distance estimation based on RSS. It has a value between [0,1] in our model for noisy RSS-based distance measurement described in equation (5).
	Total Samples Samples Per Round	The total number of beacons broadcasted by the LA. In all the simulations reported in this paper, we have used simple LA moving trajectories, in which the LA simply hovers around the sensor field and broadcasts equal number of beacons per round.
	Connectivity	The number of one-hop-neighbors of a sensor node. In simulations, the average connectivity of a network could be changed by varying the sensor radio range. Landscape-3D is independent of this parameter.

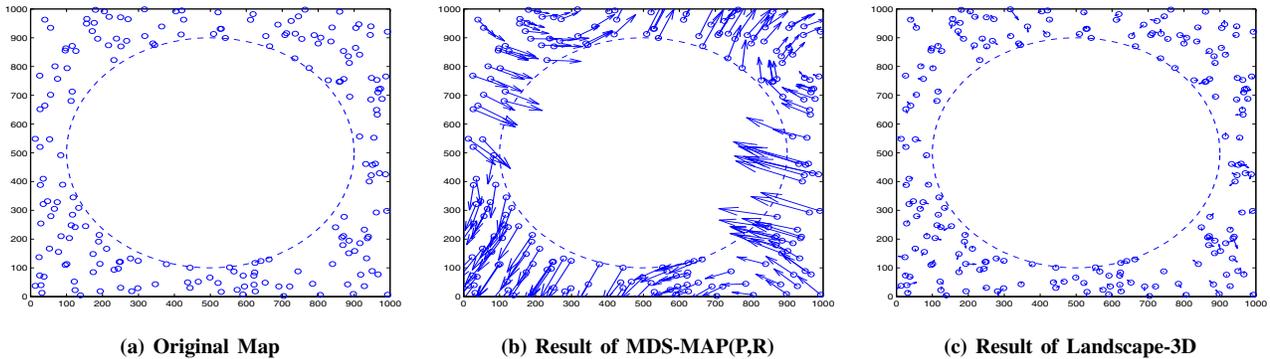


Fig. 4. Comparison of Landscape-3D and MDS-MAP for Flat terrain with irregular topology

TABLE III  
COMPARISON OF LANDSCAPE-3D AND MDS-MAP(P,R) FOR A FLAT TERRAIN WITH IRREGULAR TOPOLOGY.

	Algorithm	Parameters			Results	
		sensor radio range	connectivity	range error	position error	CPU time per node
One trail	MDS-MAP(P,R)	200	25.712	10%	106.764	0.544 sec.
	Landscape-3D	N/A	N/A	10%	11.216	0.143 sec.
Another trail	MDS-MAP(P,R)	250	33.130	10%	58.097	1.249 sec.
	Landscape-3D	N/A	N/A	10%	11.712	0.141 sec.
The average of another 1000 trails	MDS-MAP(P,R)	200	26.016	10%	115.357	0.568 sec.
	Landscape-3D	N/A	N/A	10%	11.092	0.143 sec.

the simulations reported here, we use 240 total beacon samples with 15 samples per round.

For the simulation of MDS-MAP, we have used MDS-MAP(P,R), which is the distributed version of MDS-MAP with a refinement procedure. The performance of MDS-MAP algorithm depends on the network connectivity. Generally, the higher the connectivity, the higher the accuracy and computation cost. In simulations, the connectivity could be changed by varying the sensor's radio range (Since sensor nodes are randomly deployed, even with the same sensor radio range, the connectivity could be slightly different in different trials.). In the experiments for MDS-MAP(P,R), 5% nodes are assumed as anchor nodes (with known locations).

We demonstrate the localization results of Landscape-3D and MDS(P,R) of one trail in Figure 4, in which (a) shows the original map of the sensors, (b) shows the result of MDS-

MAP(P,R), and (c) shows the result of Landscape-3D. In the figures, small circle represent the original location of sensor nodes, while small arrows point to the estimated positions. As clearly shown in the figures, MDS-MAP(P,R) does not work well for the case although the average connectivity of the network is as high as 25.7. More details of the comparison is given in Table III, in which trail one is the trail reported in Figure 4, trail two is another trail with a higher connectivity. To eliminate the effect of occasionality, the average of another 1000 trails (sensors are randomly re-deployed for each trail) is also reported in the table. Landscape-3D yields much higher accuracy with less computation overhead.

#### B. Complex Terrains with Complete Beacon Set

In this experiment, we evaluate Landscape-3D against various complex 3D terrains. Similar to the previous experiment,

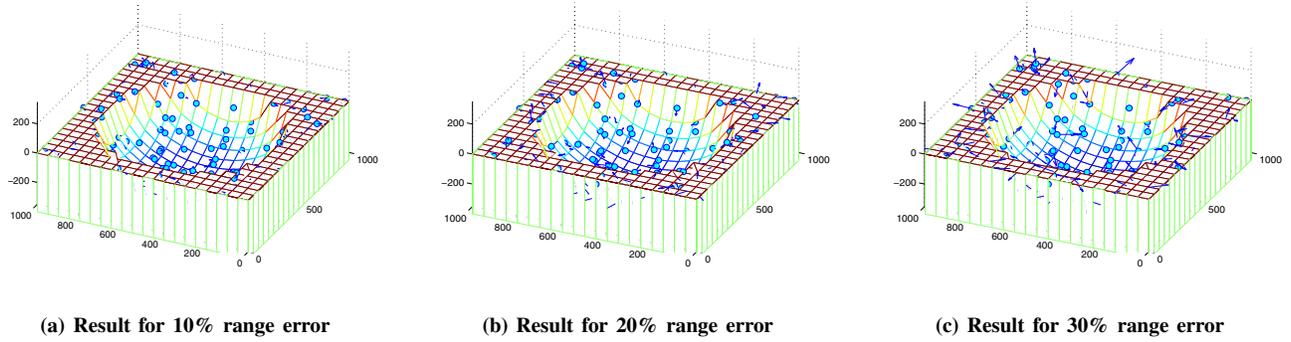


Fig. 5. The result of Landscape-3D for complex terrain 1 – Valley

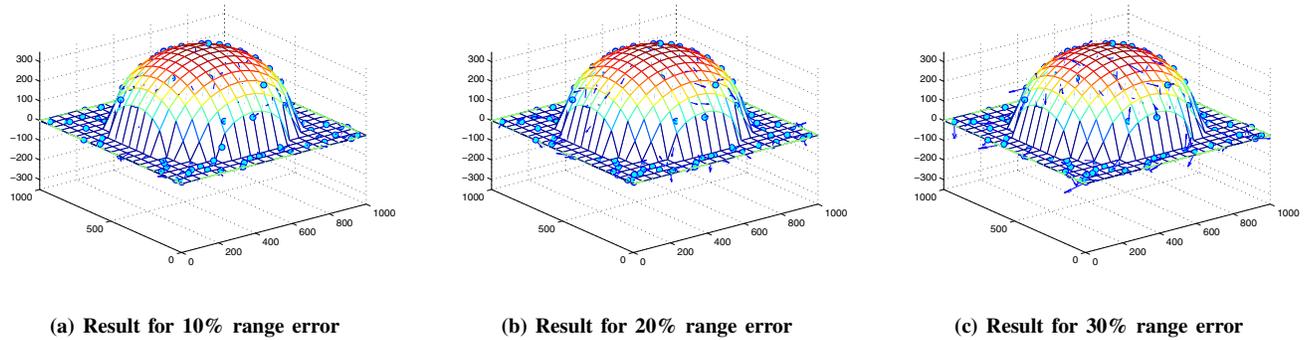


Fig. 6. The result of Landscape-3D for complex terrain 2 – Mountain

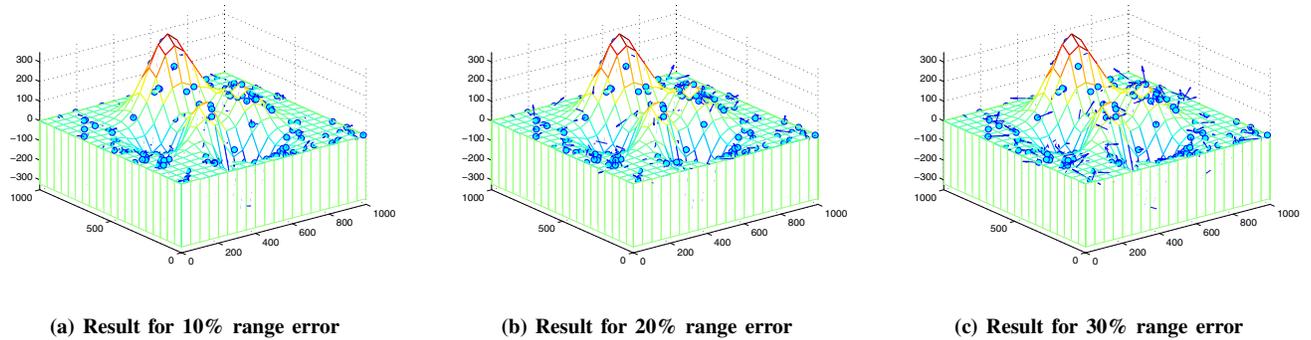


Fig. 7. The result of Landscape-3D for complex terrain 3 – Mountains and valleys

200 sensor nodes are randomly deployed over a 1000 by 1000 sensor field. The sensor field used in this experiment, however, is not a flat terrain. Three scenarios are constructed to emulate a terrain with a valley, a terrain with a mountain, and a terrain with mountains and valleys respectively. To provide references in the third dimension, we let the LA vary its height when it hovers around the sensor field. Again, a simple trajectory is employed. During the first half of the procedure (of broadcasting beacons), the LA spirals upwards. During the second half of the procedure, it spirals downwards. We use the same LA trajectory for all three scenarios (terrains). The location of the LA at time step  $n$  ( $n \geq 1$ ) could be described using the same equation as (6), except the formula for  $x_3^b(n)$  is changed as the follow:

$$x_3^b(n) = \begin{cases} c_3 & \text{if } n = 1; \\ x_3^b(n-1) + 10 & \text{if } 1 < n \leq total\_samples/2; \\ x_3^b(n-1) - 10 & \text{if } n > total\_samples/2. \end{cases}$$

In this experiment, we assume every sensor node is able to receive the complete beacon set. Figures 5, 6, and 7 demonstrate the result of Landscape-3D under three scenarios with different range errors. As shown in the figures, Landscape-3D works pretty well for all the three scenarios. Not surprisingly, we can see that the position error increases with the range error.

### C. Complex Terrains with Incomplete Beacon Set

Assuming every sensor node be able to receive all the beacons may be too optimistic. In real applications, each

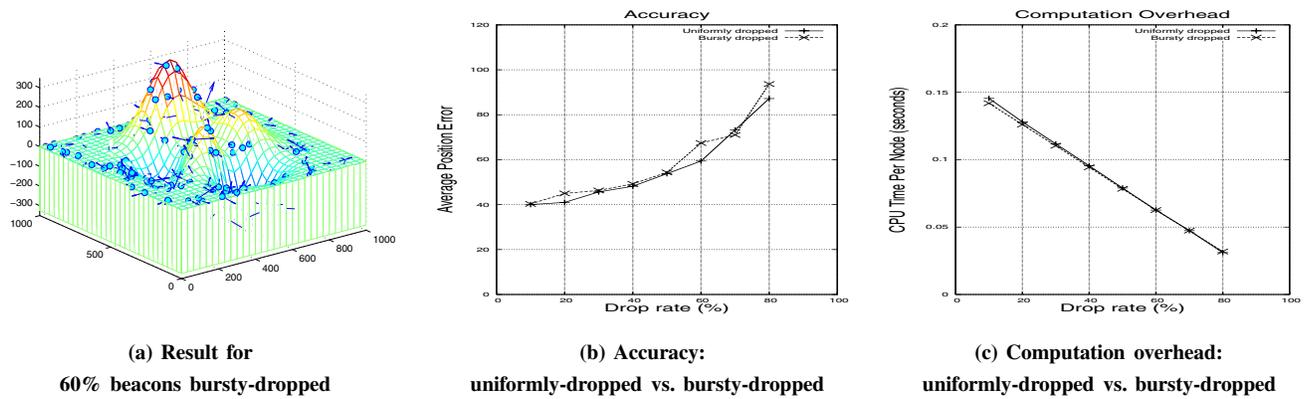


Fig. 8. The result of Landscape-3D with incomplete beacon set (with 20% range error)

sensor may individually observe a different incomplete beacon set because of obstructions, time-varying link qualities, or hostile radio-jamming attacks, etc. In this experiment, we investigate the performance of Landscape-3D against two situations of beacon loss: (a) beacons are randomly uniformly dropped; (b) beacons are randomly bursty dropped. Figure 8 demonstrates the result of the experiment, in which 20% range error is assumed. For bursty beacon drop, we set the bursty size to be a random number between 1 and 10. In Figure 8, (a) illustrates the result for a trail with 60% beacons bursty-dropped, (b) compares the accuracy for the two beacon-dropping situations under different drop rates, and (c) compares their computation costs under different drop rates. As we can see, although beacon loss does affect the accuracy of Landscape-3D, the effect is not disastrous. On the contrary, when range error is as high as 20%, Landscape-3D is still able to give acceptable result even when 60% beacons are lost. As shown in the figure, bursty-dropped beacons have slightly heavier effect than uniformly-dropped beacons. Figure 8 also clearly shows that the computation cost is linearly proportional to the number of beacons used. This provides great flexibility to sensor nodes: when the energy is low, a sensor node could intentionally drop some beacons to save the power.

## VI. CONCLUSIONS

In this paper, we have investigated a robust sensor localization scheme called Landscape-3D. Besides several advantages over existing sensor positioning approaches, such as high accuracy, high scalability, low computation cost and communication cost, one nice feature of Landscape-3D is that it works totally independent of node densities and network topologies. Landscape-3D also demonstrates strong robustness to beacon loss. Our simulation studies reveal that Landscape-3D is an effective location-finding approach for sensor networks deployed over complex 3D terrains.

## REFERENCES

- [1] A. Basu et al., "Distributed Localization by Noisy Distance and Angle Information," in *Proc. of ACM MOBIHOC'06*, May 2006.
- [2] N. Bulusu et al., "Density Adaptive Algorithms for Beacon Placement in Wireless Sensor Networks", in *Proc. of IEEE ICDCS'01*, April 2001.
- [3] W. Burgard et al., "Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach," in *Proc. of IEEE/RSJ IROS'98*, October 1998.
- [4] X. Cheng et al., "TPS, A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks", in *Proc. of IEEE INFOCOM'04*, March 2004.
- [5] K. Chintalapudi et al., "Ad-Hoc Localization Using Ranging and Sectoring", in *Proc. of IEEE INFOCOM'04*, March 2004.
- [6] L. Doherty et al., "Convex Position Estimation in Wireless Sensor Networks", in *Proc. of IEEE INFOCOM'01*, April 2001.
- [7] F. Dellaert et al., "Monte Carlo Localization for Mobile Robots", in *Proc. of ICRA'99*, May 1999.
- [8] T. Eren et al., "Rigidity, Computation, and Randomization in Network Localization", in *Proc. of IEEE INFOCOM'04*, March 2004.
- [9] T. He et al., "Range-Free Localization Schemes for Large Scale Sensor Networks", in *Proc. of ACM MOBICOM'03*, September 2003.
- [10] A. Howard et al., "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation", in *Proc. of IROS'02*, October 2002.
- [11] L. Hu and D. Evans, "Localization for Mobile Sensor Networks", to appear in *ACM MOBICOM'04*, September, 2004.
- [12] X. Ji, H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling", in *Proc. of INFOCOM'04*, March 2004.
- [13] S. Julier, "A skewed approach to filtering" in *Proc. AeroSense: Int. Symp. Aerospace/Defense Sensing, Simulation, and Controls*, Apr. 1998.
- [14] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *Proc. of the IEEE*, Vol. 92, No. 3, March 2004.
- [15] C. Kwok et al., "Adaptive Real-time Particle Filters for Robot Localization", in *Proc. of ICRA'03*, September 2003.
- [16] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", in *Proc. of ACM MOBICOM'00*, August, 2000.
- [17] R. Merwe et al., "The unscented particle filter". *Technical Report CUED/F-INFENG/TR 380*, Cambridge University Engineering Department, Aug. 2000.
- [18] N.B. Priyantha et al., "Anchor-Free Distributed Localization in Sensor Networks", *LCS Technical Report #892*, MIT, April 2003.
- [19] A. Savvides et al., "The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems", in *Proc. of ACM WSNA'02*, September 2002.
- [20] Y. Shang et al., "Localization From Mere Connectivity", in *Proc. of ACM MOBIHOC'03*, June 2003.
- [21] Y. Shang, W. Ruml, "Improved MDS-Based Localization", in *Proc. of IEEE INFOCOM'04*, March 2004.
- [22] I. Stojmenovic, "Location updates for efficient routing in ad hoc wireless networks", *Handbook of Wireless Networks and Mobile Computing*, Wiley, 451-471, 2002.
- [23] K. Whitehouse and D. Culler, "Calibration as Parameter Estimation in Sensor Networks", in *Proc. of ACM WSNA'02*, September 2002.
- [24] K. Whitehouse, "Calamari: a sensor field localization system", <http://www.cs.berkeley.edu/~kamin/calamari>.
- [25] L. Zhang et al., "Landscape: A High Performance Distributed Positioning Scheme for Outdoor Sensor Networks", in *Proc. of IEEE WiMob 2005*, August 2005.
- [26] L. Zhang et al., "A Novel Distributed Sensor Positioning System Using the Dual of Target Tracking", *online technical report*, available at <http://www.cs.iusb.edu/~liqzhang/Liqiang-Landscape.pdf>.