Robust Localization from Incomplete Local Information

Amin Karbasi, *Student Member, IEEE*, Sewoong Oh, *Member, IEEE*

***Abstract*—We consider the problem of localizing wireless devices in an ad-hoc network embedded in a d-dimensional Euclidean space. Obtaining a good estimate of where wireless devices are located is crucial in wireless network applications including environment monitoring, geographic routing and topology control. When the positions of the devices are unknown and only local distance information is given, we need to infer the positions from these local distance measurements. This problem is particularly challenging when we only have access to measurements that have limited accuracy and are incomplete. We consider the extreme case of this limitation on the available information, namely only the connectivity information is available, i.e., we only know whether a pair of nodes is within a fixed detection range of each other or not, and no information is known about how far apart they are. Further, to account for detection failures, we assume that even if a pair of devices is within the detection range, it fails to detect the presence of one another with some probability and this probability of failure depends on how far apart those devices are. Given this limited information, we investigate the performance of a centralized positioning algorithm** MDS-MAP **introduced by Shang et al. [3], and a distributed positioning algorithm** HOP-TERRAIN **introduced by Savarese et al. [4]. In particular, for a network consisting of n devices positioned randomly, we provide a bound on the resulting error for both algorithms. We show that the error is bounded, decreasing at a rate that is proportional to R_{Critical}/R, where R_{Critical} is the critical detection range when the resulting random network starts to be connected, and R is the detection range of each device.**

***Index Terms*—centralized, distributed, localization, sensor network**

I. Introduction

In this paper, we address the problem of positioning when only a partial information on pairwise distances is provided. Location estimation of individual nodes is required for many wireless sensor network applications such as environment monitoring, geographic routing and topology control, to name only a few [5], [6]. In environment monitoring, for instance, the environmental measurement data by the wireless sensor network is more useful when accompanied by the location information. One way to acquire the positions is to equip all the sensors with a global positioning system (GPS). The use of GPS not only adds considerable cost to the system, but more importantly, it does not work in indoor environments or when the received GPS signal is jammed [7]. Alternatively, we need an algorithm that can derive positions of sensors based on local and basic information such as proximity (which nodes are within communication range of each other) or local distances (pairwise distances between neighbouring sensors).

Two common techniques for obtaining the local distance and connectivity information are Received Signal Strength Indicator (RSSI) and Time Difference of Arrival (TDoA). RSSI is a measurement of the ratio of the power present in a received radio signal and a reference power. Signal power at the receiving end decreases as a function of the distance, and RSSI has the potential to be used to estimate the distance. Alternatively, TDoA techniques use the time difference between the receipt of two different signals with different velocities, for instance ultrasound and radio frequency signals [8], [9]. These techniques can be used, independently or together, for distance estimation.

Given a set of such measurements, we want to find the positions. One common approach, known as multi-dimensional scaling (MDS) [10], assumes that all pairwise distances are known. However, in almost all practical scenarios such information is unavailable for two major reasons. First, sensors are typically highly resource-constrained (e.g., power) and have limited communication range. Thus, far away sensors cannot communicate and obtain their pairwise distances. Second, due to noise and interference among sensors, there is always the possibility of non-detection or completely incoherent measurements.

Many algorithms have been proposed to resolve these issues by using heuristic approximations to the missing distances, and their success has mostly been measured experimentally. Regarding the mechanisms deployed for estimating sensor locations, one can divide the localization algorithms into two categories: range-based and range-free. In the range-based protocols the absolute point-to-point distance estimates are used for inferring the locations, whereas in the range-free protocols no assumptions about the availability of such information are made and only the connectivity information is provided. As a result, range-free algorithms are more effective in terms of stability and cost, hence more favourable to be deployed in practical settings. In sensor network literature, "range-free" is also referred to as "connectivity-based", and "range-based" is also referred to as "range-aware".

The theoretical guarantees associated with the performance of the existing methods are, however, of the same interest and complementary in nature. Such analytical bounds on the performance of localization algorithms can provide answers to practical questions: for example, "How large should the radio range be in order to get the reconstruction error within

Amin Karbasi is with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (email: amin.karbasi@epfl.ch). Sewoong Oh is with the department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (email: swoh@illinois.edu). A preliminary summary of this work appeared in [1], [2].

a threshold?" With this motivation in mind, our work takes a step forward in this direction.

We first focus on providing a bound on the performance of a popular localization algorithm MDS-MAP [3] when applied to sensor localization from only connectivity information. We should stress here that pairwise distances are invariant under rigid transformations (rotation, translation and reflection). Hence, given connectivity information, we can only hope to determine the *configuration* or the relative map of the sensors. In other words, localization is possible only up to rigid transformations. With this point in mind, we prove that using MDS-MAP, we are able to localize sensors up to a bounded error in a connected network where most of distances are missing and only local connectivity information is given.

More precisely, assume that there are n sensors positioned randomly in a d-dimensional unit cube with the radio range $R = o(1)$. Further, assume that each pair can detect each other with probability at least p_0. Let the $n \times d$ matrices X and $\hat{X}$ denote the true sensor positions and their estimates by MDS-MAP, respectively. Define $L = \mathbb{I}_{n\times n} - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ where $\mathbb{I}_{n\times n}$ is the identity matrix and $\mathbb{1}_n$ is the all ones vector. The quantity LXX^TL satisfies nice properties: (a) it is invariant under rigid transformations and (b) if $LXX^TL = L\hat{X}\hat{X}^TL$, then X and $\hat{X}$ are equal up to rigid transformations. Therefore, a natural distance metric is:

$$d_{\text{inv}}(X, \widehat{X}) = \frac{1}{n}\left\|LXX^TL - L\widehat{X}\widehat{X}^TL\right\|_F,$$

where $\|\cdot\|_F$ denote the Frobenius norm. Using this, we establish an upper bound on the error of MDS-MAP:

$$d_{\text{inv}}(X, \widehat{X}) \leq \frac{R_{\text{MDS}}}{R} + o(1),$$

where $R_{\text{MDS}} = C_d(\ln(n)/(p_0\, n))^{1/d}$ for some constant C_d that only depends on the dimension d.

One consequence of the ad-hoc nature of the underlying networks is the lack of a central infrastructure. This hinders the use of a centralized algorithm like MDS-MAP. In particular, centralized algorithms suffer from scalability problems, making it difficult to implement in large scale sensor networks. Centralized algorithms also require higher computational complexity [11]. This leads us to investigate if similar performance guarantees can be obtained in a distributed setting, where each sensor tries to estimate its own *global* position. As mentioned above, this task cannot be accomplished unless some additional information, other than local measurements, is provided. It is well known that in a d-dimensional Euclidean space, we need to know the global positions of at least $d+1$ sensors, referred to as *anchors*, in order to uniquely determine the global positions [12].

Under such decentralized scenario, we analyze the performance of a popular localization algorithm: HOP-TERRAIN [4]. This algorithm can be seen as a distributed version of the MDS-MAP. We prove that HOP-TERRAIN can localize sensors up to a bounded error in a connected network where most of the pairwise distances are unknown and only local connectivity information is given.

Similarly as in the case of MDS-MAP, assume n sensors in a d-dimensional unit cube and $d+1$ anchors in general positions. We show that when only connectivity information is available, the Euclidean distance between the estimate $\hat{x}_i$ and the correct position x_i is bounded by

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + o(1),$$

for all i where $R_{\text{HOP}} = C'_d(\log n/(p_0 n))^{\frac{1}{d}}$ for some constant C'_d that only depends on d.

Preliminary results on analyzing the performance of MDS-MAP and HOP-TERRAIN was presented at SIGMETRICS 2010 and Information Theory Workshop in 2010 [1], [2]. The main new contribution of this paper is that we generalize the previous results by considering a more general scenario where two sensors within a communication radio range can still fail to detect the presence of each other due to hostile environment or sensor malfunction. In particular, the main results presented in this paper includes as a special case the results in both of the previous conference papers under traditional disc model. Further, we provide a complete proofs of these results, which was not present in previous papers.

II. Related work

The localization problem has attracted significant research interests in recent years. A general survey of the area and an overview of recent techniques can be found in [12] and [11]. In the case when all pairwise distances are known, the coordinates can be derived by using a classical method known as multidimensional scaling (MDS) [10]. The underlying principle of the MDS is to convert distances into an inner product matrix, whose singular vectors are the unknown coordinates. In the presence of noise, MDS tolerates errors gracefully due to the overdetermined nature of the problem. However, when most distances are missing, finding coordinates becomes more challenging. Three types of practical centralized algorithms have been proposed in the literature. The first group consists of algorithms that try first to estimate the missing entries of the distance matrix and then apply MDS to the reconstructed distance matrix to find the coordinates of the sensors. MDS-MAP, introduced in [3] and further studied in [14], is a well-known example, where it computes the shortest paths in order to approximate the missing distances. The algorithms in the second group mainly consider the sensor localization as a non-convex optimization problem and directly estimate the coordinates of sensors. A famous example of this type is a relaxation to semidefinite programming (SDP)[15]. In the third group, the problem is formulated through a stochastic optimization where the main technique used in these algorithms is the stimulated annealing, which is a generalization of the Monte Carlo method in combinatorial optimization [16], [17].

Perhaps a more practical and interesting case is when there is no central infrastructure. [13] identifies a common three-phase structure of three popular distributed sensor-localization algorithms, namely robust positioning [4], ad-hoc positioning [18] and N-hop multilateration [19]. Table I illustrates the structure of these algorithms. In the first phase, nodes share information to collectively determine the distances from each of the nodes to a number of anchors. Anchors are special

TABLE I
DISTRIBUTED LOCALIZATION ALGORITHM CLASSIFICATION [13]

Phase	Robust positioning	Ad-hoc positioning	N-hop multilateration
1. Distance	DV-HOP	Euclidean	Sum-dist
2. Position	Lateration	Lateration	Min-max
3. Refinement	Yes	No	Yes

nodes with a priori knowledge of their own position. In the second phase, nodes determine their position based on the estimated distances to the anchors. In the last phase, the initial estimated positions are iteratively refined. It is empirically demonstrated that these simple three-phase distributed sensor-localization algorithms are robust and energy-efficient [13]. However, depending on which method is used in each phase, there are different trade-offs between localization accuracy, computation complexity and power requirements.

The performances of these algorithms are measured through simulations and little is known about their theoretical analysis. A few exceptions are in the following work. In [20] the authors use matrix completion methods [21] as a means to reconstruct the distance matrix. The main contribution of their paper is that they are able to provably localize the sensors up to a bounded error. However, their analysis is based on a number of strong assumptions. First, they assume that even far-away sensors have a non-zero probability of detecting their distances. Second, the algorithm explicitly requires the knowledge of detection probabilities between all pairs. Third, their theorem only works when the average degree of the network (i.e., the average number of nodes detected by each sensor) grows linearly with the number of sensors in the network.

Our first result on the analysis of MDS-MAP provides a theoretical guarantee that backs up experimental results. We use shortest paths as our primary guess for the missing entries in the distance matrix and apply MDS to find the relative positions of the nodes up to a rigid motion. In contrast to [20], we require significantly weaker assumptions. More specifically, we assume that only neighbouring sensors have information about each other and that only connectivity information is known. Furthermore, for the purpose of estimating the positions, the algorithms presented in this paper do not require the knowledge of the detection probability. And last, in our analysis we assume that the average degree grows logarithmically, instead of linearly, with the number of sensors, which results in needing many less revealed entries in the distance matrix. On one hand, we would like to choose the radio range large enough such that the graph is connected. Otherwise, we would be wasting the part of the graph that is not connected to the giant component. On the other hand, we would like to use as small a radio range as possible to save power consumption and operation cost. Hence, choosing a radio range which gives the average number of neighbors of order $\log n$ is desired. We provide the first error bounds on the performance of MDS-MAP.

Of particular interest are the two new results on the performance of sensor localization algorithms. In [22], Javanmard et al. proposes a new reconstruction algorithm based on semidefinite programming where they could establish lower and upper bounds on the reconstruction errors of their algorithm. Similarly, in [23], due to new advances in matrix completion methods [24], the authors analyze the performance of OptSpace [25], a novel matrix completion algorithm, in localizing the sensors. Interestingly, they did not need to adhere to the assumptions made by [20]. However, they have a restrictive assumption about the topology of the network: sensors are scattered inside an annulus.

The above analytical results crucially rely on the fact that there is a central processor with access to the distance measurements. However, centralized algorithms suffer from the scalability problem and require higher computational power. Hence, a distributed algorithm with similar a performance bound is desirable. In our second result, we analyze the reconstruction error of a distributed algorithm. To the best of our knowledge, we show for the first time that HOP-TERRAIN, introduced in [4], achieves a bounded error when only local connectivity information is given.

The organization of this paper is as follows. In Section III we introduce the model used in our work. In Section IV we describe the MDS-MAP and HOP-TERRAIN algorithms. Our main results are stated in Section V, and we provide the proofs in Section VI. Finally, we conclude in Section VII.

III. MODEL DEFINITION

In this section, we define a probabilistic model considered in this work. We assume we have no fine control over the placement of the sensors that we call *unknown nodes* (e.g., the nodes are dropped from an airplane). Hence, n nodes are placed uniformly at random in a d-dimensional cube $[0,1]^d$. We assume that there are m special sensors, which we call *anchors*, with a priori knowledge of their own positions in some global coordinate. In practice, it is reasonable to assume that we have some control over the position of anchors. Anchors can be some nodes that are planted on the field before any positioning takes place. Let $V_a = \{1,\dots,m\}$ denote the set of m vertices corresponding to the anchors and $V_u = \{m+1,\dots,m+n\}$ the set of n vertices corresponding to the unknown nodes. We use x_i to denote the position of node i and $X \in \mathbb{R}^{n\times d}$ to denote the position matrix where the i-th row corresponds to x_i.

In positioning, due to attenuation and power constraints, only measurements between close-by nodes are available. As a result, the pairwise distance measurements can be represented by a random geometric graph $G(n+m,R) = (V,E,P)$, where $V = V_u \cup V_a$, $E \subseteq V \times V$ is a set of edges that connect pairs of sensors that are detected, and $P : E \to \mathbb{R}^+$ is a non-negative real-valued function. Edge weights P is a mapping from an edge (i,j) to a pairwise distance measurement.

A common model for this random geometric graph is the disc model where node i and j are connected if the Euclidean distance $d_{i,j} \equiv \|x_i - x_j\|$ is at most a positive radio range

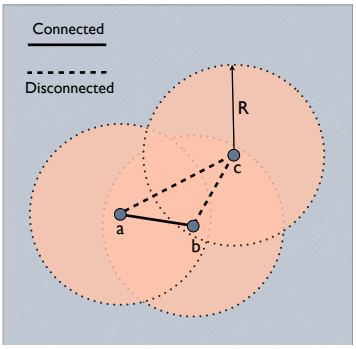

Fig. 1. This example shows the model. Nodes a and b are connected since they are within R. Although b and c are also within R, they are not connected due to detection failure. a and c are not connected because they are far apart.

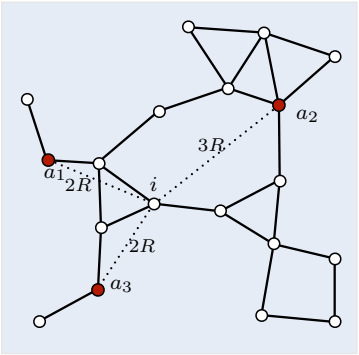

Fig. 2. The shortest path between two nodes is defined in terms of the minimum number of hops multiplied by the radio range R.

R. There are a variety of ways to measure the connectivity between two nodes, including TDoA and RSSI. Due to limited resources, there is a probability of non-detection. Think of RF ranging in the presence of an obstacle or in the case of multiple paths. Depending on the acquisition mechanism, this may result in the absence of measurements.

To model this failure of detection, we assume that two nodes can detect each other with a probability that only depends on the distance $d_{i,j}$. Namely, $(i,j) \in E$ with probability $p(d_{i,j})$ if $d_{i,j} \leq R$. The detection probability $p(\cdot) : [0,R] \to [0,1]$ is a non-increasing function of the distance. Our main results on the error achieved by localization algorithms assumes that the detection probability $p(\cdot)$ is lower bounded by a parameter p_0 such that

$$p(z) \geq p_0 \ ,$$

for $z \leq R$. For example, $p(\cdot)$ might be a simple function parameterized by two scalar values p_0 and β:

$$p(z) = \min\left(1, p_0 \left(\frac{z}{R}\right)^{-\beta}\right) \ , \tag{1}$$

for some $p_0 \in (0,1]$ and β. This includes the disc model with perfect detection as a special case (i.e., $p_0 = 1, \beta = 0$). An example is shown in Figure 1.

To each edge $(i,j) \in E$, we associate the distance measurement $P_{i,j}$ between sensors i and j. In an ideal case, we have exact distance measurements available for those pairs in E. This is called the *range-based model*. Formally, $P_{i,j} = d_{i,j}$ if $(i,j) \in E$ and $*$ otherwise, where a $*$ denotes that the distance measurement is unavailable. In the following, the algorithms presented in this paper only uses values of $P_{i,j}$'s where it is well defined (i.e. for $(i,j) \in E$, and equivalently for $P_{i,j} \neq *$). In this paper, we assume that we are given only network connectivity information and no distance information. This is known as the *connectivity-based model*. Formally,

$$P_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ * & \text{otherwise.} \end{cases}$$

In the following, let D denote the $n \times n$ squared distance matrix where $D_{i,j} = d_{i,j}^2$. By definition, $D = a\mathbb{1}_n^T + \mathbb{1}_n a^T - 2XX^T$, where $a \in \mathbb{R}^n$ is a vector with $a_i = \|x_i\|^2$ and $\mathbb{1}_n$ is the all ones vector. As D is a sum of two rank-1 matrices and a rank-d matrix, its rank is at most $d+2$. A summary of notation is provided in Table II.

IV. Algorithms

Depending on the application, we might want a *relative map* or an *absolute map* of the locations. A relative map is a configuration that have the same neighborhood relationships as the underlying graph G. In the following we use the terms configuration and relative map interchangeably. An absolute map, on the other hand, determines the absolute geographic coordinates. In this paper, our objective is two-fold. First, we present a centralized algorithm MDS-MAP, that finds a configuration that best fits the proximity measurements. Then, we discuss its distributed version, HOP-TERRAIN, where the goal is for each sensor to find its absolute position.

A. *Centralized Positioning Algorithm:* MDS-MAP

The centralized MDS-MAP algorithm, assumes no anchors in the system. We denote the random positions of n sensors by x_i's. MDS-MAP consists of two steps:

Algorithm 1 MDS-MAP [3].

Input: dimension d, graph $G = (V, E, P)$.

1: Compute the shortest paths, and let $\widehat{D}$ be the squared shortest paths matrix.
2: Apply MDS to $\widehat{D}$, and let $\widehat{X}$ be the output.

Shortest paths. The shortest path between nodes i and j in graph $G = (V, E, P)$ is defined as a path between two nodes such that the sum of the proximity measures of its constituent edges is minimized (see Figure 2). Let $\hat{d}_{i,j}$ be the computed shortest path between node i and j. Then, the squared shortest paths matrix $\widehat{D} \in \mathbb{R}^{n \times n}$ is defined as $\widehat{D}_{ij} = \hat{d}_{i,j}^2$ for $i \neq j$, and 0 for $i = j$.

Multidimensional scaling. In step 2, we apply the Multi-dimensional scaling (MDS) to $\widehat{D}$ to get a good estimate of X, specifically, we compute $\widehat{X} = \mathsf{MDS}_d(\widehat{D})$. MDS refers to a set of statistical techniques used in finding the configuration of objects in a low dimensional space such that the measured pairwise distances are preserved [10]. It is often used for a

TABLE II
SUMMARY OF NOTATION.

n	number of unknown sensors	V_u	set of unknown nodes
m	number of anchors	V_a	set of anchors
R	communication range	$\mathbb{1}_n$	all ones vector of size n
$P_{i,j}$	distance measurements	$\widehat{D}$	estimated squared distance matrix
$d_{i,j}$	Pairwise distance between nodes i and j	$\mathbb{I}_{n\times n}$	$n \times n$ identity matrix
x_i	position of node i	$\hat{x}_i$	estimated position of node i
p_0	minimum detection probability	X	positions matrix
d	dimension	$\widehat{X}$	estimated positions matrix
D	squared distance matrix	$\hat{d}_{i,j}$	shortest path between node i and j
$\mathsf{O}(d)$	orthogonal group of $d \times d$ matrices	$\left\|\cdot\right\|_F$	Frobenius norm
$\langle A, B\rangle$	Frobenius inner product	$\left\|\cdot\right\|_2$	spectral norm

visual representation of the proximities between a set of items. Formally, MDS finds a lower dimensional embedding $\hat{x_i}$s that minimize the *stress*:

$$\text{stress} \equiv \sqrt{\frac{\sum_{i\neq j}(f(d_{i,j}) - \hat{d}_{i,j})^2}{\sum_{i\neq j}\hat{d}_{i,j}^2}} ,$$

where $d_{i,j}$ is the input similarity (or dissimilarity), $\hat{d}_{i,j} = \|\hat{x_i} - \hat{x_j}\|$ is the Euclidean distance in the lower dimensional embedding, and $f(\cdot)$ is some function on the input data. When MDS perfectly embeds the input data, we will have $f(d_{i,j}) = \hat{d}_{i,j}$ and the stress is zero. In this section, we use

Algorithm 2 Classic Metric MDS [3].

Input: Dimension d, estimated squared distance matrix $\widehat{D}$
Output: Estimated positions $\mathsf{MDS}_d(\widehat{D})$
1: Compute $(-1/2)LML$, where $L = \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$.
2: Compute the best rank-d approximation $U_d\Sigma_d U_d^T$ of $(-1/2)LML$.
3: Return $\mathsf{MDS}_d(M) \equiv U_d\Sigma_d^{1/2}$.

what is called the classic metric MDS [26], where $f(\cdot)$ is the identity function and the input dissimilarities correspond to the Euclidean distances such that $d_{i,j} = \|x_i - x_j\|$ for some lower dimensional embedding $\{x_i\}$. This algorithm has been frequently used in positioning applications; and from here on whenever we say MDS we refer to the above algorithm. Let $\mathsf{MDS}_d(D)$ denote the $n \times d$ matrix returned by MDS when applied to the squared distance matrix D. Then, in formula, given the singular value decomposition (SVD) of a symmetric and positive definite matrix $(-1/2)LDL$ as $(-1/2)LDL = U\Sigma U^T$,

$$\mathsf{MDS}_d(D) \equiv U_d\Sigma_d^{1/2} ,$$

where U_d denotes the $n \times d$ left singular matrix that corresponds to the first d singular values and Σ_d denotes the $d \times d$ diagonal matrix with top d singular values. This is also known as the MDSLOCALIZE algorithm in [20]. Note that as the columns of U are orthogonal to $\mathbb{1}_n$ by construction, it follow that $L \cdot \mathsf{MDS}_d(D) = \mathsf{MDS}_d(D)$.

It is crucial that we apply the double scaling by L to the distance matrix:

$$D = a\mathbb{1}_n^T + \mathbb{1}_n a^T - 2XX^T .$$

Since by construction, L is orthogonal to $\mathbb{1}_n$, but preserves the $n-1$ dimensional complementary subspace, the double scaling eliminates the first two terms. Hence, when MDS is applied to D without noise, the configuration of sensors are exactly recovered up to a rigid motion:

$$-(1/2)LDL = LXX^TL . \tag{2}$$

Note that we only obtain the configuration and not the absolute positions, in the sense that $\mathsf{MDS}_d(D)$ is one version of infinitely many solutions that matches the distance measurements D. We introduce a formal definition of rigid transformation and related terms below.

Let $\mathsf{O}(d) = \{Q|QQ^T = Q^TQ = \mathbb{I}_d\}$ denote the orthogonal group of $d \times d$ matrices. We say $Y \in \mathbb{R}^{n\times d}$ is a rigid transformation of X, if there exists a shift vector $s \in \mathbb{R}^d$ and an orthogonal matrix $Q \in \mathsf{O}(d)$ such that $Y = XQ + \mathbb{1}_n s^T$. Here Y is a result of rotating X by Q and then adding a shift by s. Similarly, when we say two position matrices X and Y are equal up to a rigid transformation, we mean that there exists a rotation Q and a shift s such that $Y = XQ + \mathbb{1}_n s^T$. Also, we say a function $f(X)$ is *invariant* under rigid transformation if and only if for all X and Y that are equal up to a rigid transformation we have $f(X) = f(Y)$. Under these definitions, it is clear that D is invariant under rigid transformation, as for all (i, j), since $D_{ij} = \|x_i - x_j\|^2 = \|(x_iQ + s^T) - (x_jQ + s^T)\|^2$, for any $Q \in \mathsf{O}(d)$ and $s \in \mathbb{R}^d$.

Although MDS works perfectly when D is available, in practice not all proximity measurements are available because of the limited radio range R. This is why, in the first step, we estimated the unavailable entries of D by finding the shortest path between disconnected nodes.

B. Distributed Positioning Algorithm: HOP-TERRAIN

Recall that HOP-TERRAIN is a distributed algorithm that aims at finding the global map. In order to fix the global coordinate system in a d dimensional space, we need to know the positions of at least $d + 1$ anchors, nodes with known positions. In this section, we assume that we have m anchors. Based on the robust positioning algorithm introduced in [4], the distributed sensor localization algorithm consists of two steps:

Distributed shortest paths: Similar to MDS-MAP, the first step is about finding the shortest path. The difference is that in the first step each of the unknown nodes only estimates the distances between itself and the anchors. These approximate distances will be used in the next triangulation step to derive an estimated position. In other words, the

Algorithm 3 MDS-MAP [3].

1: Each node i computes the shortest paths $\{\hat{d}_{i,a} : a \in V_a\}$ between itself and the anchors.
2: Each node i derives an estimated position $\hat{x}_i$ by triangulation with a least squares method.

shortest path between an unknown node i and an anchor a in the graph G provides an estimate for the Euclidean distance $d_{i,a} = \|x_i - x_a\|$. We denote by $\hat{d}_{i,a}$ the computed length of the shortest path. When the corresponding graph is defined as in the *connectivity-based model*, the shortest path $\hat{d}_{i,a}$ is equivalent to the minimum number of hops between two nodes, scaled by the radio range R.

In order to compute the number of hops in a distributed way, we use a method similar to DV-HOP [18]. Each unknown node maintains a table $\{x_a, h_a\}$ that is initially empty, where $x_a \in \mathbb{R}^d$ refers to the position of the anchor a and h_a to the number of hops from the unknown node to the anchor a. First, each of the anchors initiate a broadcast containing its known location and a hop count of one. All of the one-hop neighbors surrounding the anchor, on receiving this broadcast, record the anchor's position and a hop count of one, and then broadcast the anchor's known position and a hop count of two. From then on, whenever a node receives a broadcast, it does one of the two things. If the broadcast refers to an anchor that is already in the record and the hop count is larger than or equal to what is recorded, then the node does nothing. Otherwise, if the broadcast refers to an anchor that is new or has a hop count that is smaller, the node updates its table with this new information on its memory and broadcasts the new information after incrementing the hop count by one. When every node has computed the hop count to all the anchors, the number of hops is multiplied by the radio range R. Note that to begin triangulation, not all the hop counts to all the anchors are necessary. A node can start triangulation as soon as it has estimated distances to $d+1$ anchors.

The above step of computing the minimum number of hops is the same distributed algorithm as described in DV-HOP. However, one difference is that instead of multiplying the number of hops by a fixed radio range R, in DV-HOP, the number of hops is multiplied by an average hop distance. The average hop distance is computed from the known pairwise distances between anchors and the number of hops between the anchors. Although numerical simulations show that the average hop distance provides a better estimate, the difference between the computed average hop distance and the radio range R becomes negligible as n grows large.

Triangulation using least squares. In the second step, each unknown node i uses a set of estimated distances $\{\hat{d}_{i,a} : a \in V_a\}$ together with the known positions of the anchors, to perform a triangulation. The resulting estimated position is denoted by $\hat{x}_i$. For each node, the triangulation consists in solving a single instance of a least squares problem ($Ax = b$) and this process is known as Lateration [27], [13]. The position vector x_i and the anchor positions $\{x_a : a \in \{1, \ldots, m\}\}$ satisfy the following:

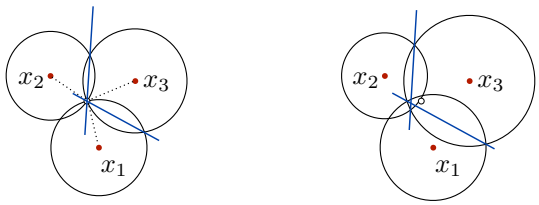

Fig. 3. Multilateration with exact distance measurements (top) and with approximate distance measurements (bottom). Three solid circles denote the anchors (red) and the white circle denotes the unknown nodes. The intersection of the blue lines corresponds to the solution of multilateration.

$$\begin{aligned} \|x_1 - x_i\|^2 &= d_{i,1}^2 \,, \\ &\vdots \\ \|x_m - x_i\|^2 &= d_{i,m}^2 \,. \end{aligned}$$

Geometrically, the above equalities simply say that the point x_i is the intersection point of m circles centred at $x_1, x_2, \ldots, x_m$ (see Figure 3). This set of equations can be linearized by subtracting each line from the next line. By reordering the terms, we get a series of linear equations in the form $A\,x_i = b_0^{(i)}$, for $A \in \mathbb{R}^{(m-1)\times d}$ and $b \in \mathbb{R}^{m-1}$ defined as

$$\begin{aligned} A &\equiv \begin{bmatrix} 2(x_1 - x_2)^T \\ \vdots \\ 2(x_{m-1} - x_m)^T \end{bmatrix} , \\ b_0^{(i)} &\equiv \begin{bmatrix} \|x_1\|^2 - \|x_2\|^2 + d_{i,2}^2 - d_{i,1}^2 \\ \vdots \\ \|x_{m-1}\|^2 - \|x_m\|^2 + d_{i,m}^2 - d_{i,m-1}^2 \end{bmatrix} . \end{aligned}$$

Note that the matrix A does not depend on the particular unknown node i and all the entries are known accurately to all the nodes. However, the vector $b_0^{(i)}$ is not available at node i, because $d_{i,a}$'s are not known. Hence we use an estimation $b^{(i)}$, that is defined from $b_0^{(i)}$ by replacing $d_{i,a}$ by $\hat{d}_{i,a}$ everywhere. Notice that $\hat{d}_{i,a} \geq d_{i,a}$. As a result, the circles centred at $x_1, x_2, \ldots, x_m$ have potentially larger radii. Therefore, the intersection between circles is no longer a single point, but rather a closed area. Then, finding the optimal estimation $\hat{x}_i$ of x_i that minimizes the mean squared error is solved in a closed form using a standard least squares approach:

$$\hat{x}_i = (A^T A)^{-1} A^T b^{(i)} \,. \tag{3}$$

C. Computational complexity and network connectivity

Complexity analysis. For bounded $d = o(1)$, a single least squares operation has complexity $O(m)$, and applying it n times results in the overall complexity of $O(n\,m)$. No communication between the nodes is necessary for this step. In the MDS-MAP algorithm we require that all-pairs shortest paths be found. This problem has an efficient algorithm whose complexity is $O(n^2 \log n + n|E|)$ [28]. For $R = C(\log n/n)^{1/d}$ with constant C, the graph is sparse with $|E| = O(n \log n)$,

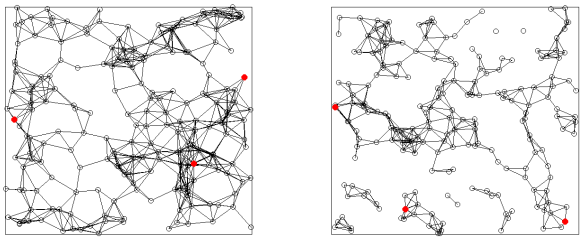

Fig. 4. The red vertices indicate the anchors. Under the right scaling of the radio range R, the graph stays connected (top figure) whereas otherwise there will be nodes without any means of communication to others (bottom graph).

whence the complexity is $O(n^2 \log n)$. Contrary to MDS-MAP, in HOP-TERRAIN we must only compute the shortest paths between the unknown nodes and the anchors. This distributed shortest paths algorithm can be done efficiently with total complexity of $O(n\,m)$.

Network connectivity. In general when the graph G is not connected, the localization problem is not well defined, and there are multiple configurations resulting in the same observed proximity measures. For instance if graph G consists of two disconnected components, they can be placed in any way with respect to each other without violating any constraints imposed by G.

In this work, we are interested in a scalable system of n unknown nodes for a large value of n. As n grows, it is reasonable to assume that the average number of connected neighbors for each node should stay constant. This happens, in our model, if we chose the radio range $R = C/n^{1/d}$. However, in the unit square, assuming sensor positions are drawn uniformly, the random geometric graph is connected, with high probability, if $\pi R^2 > (\log n + c_n)/n$ for $c_n \to \infty$ [29]. A similar condition can be derived for generic d-dimensions as $C_d R^d > (\log n + c_n)/n$, where $C_d \leq \pi$ is a constant that depends on d. Moreover, in case $C_d R^d < (\log n + c_n)/n$, not only the graph is not connected, there will be *isolated nodes* with hight probability. In this case, both MDS-MAP and HOP-TERRAIN algorithms will be in trouble (see Figure 4). Hence, we focus in the regime where the average number of connected neighbors is slowly increasing with n. Let R_{critical} be the critical detection range where the resulting graph starts to be connected. Then we are interested in the regime $R = CR_{\text{Critical}}$, for some positive constant $C \geq 1$ such that the graph stays connected with high probability.

V. Main results

In this section we present our main results regarding the performance of MDS-MAP and HOP-TERRAIN algorithms.

A. MDS-MAP

Our first result establishes an upper bound on the error achieved by MDS-MAP under *connectivity-based model*. Let $\widehat{X}$ denote an estimate for X. Then, we need to define a metric for the distance between the original position matrix X and the estimation $\widehat{X}$, which is invariant under rigid transformation. Define $L \equiv \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ as in the MDS algorithm. This naturally defines a distance between X and $\widehat{X}$:

$$d_{\text{inv}}(X, \widehat{X}) = \frac{1}{n}\left\| LXX^TL - L\widehat{X}\widehat{X}^TL \right\|_F , \tag{4}$$

where $\|A\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ denotes the Frobenius norm. Notice that the factor $(1/n)$ corresponds to the usual normalization by the number of entries in the summation. Indeed this distance is invariant to rigid transformation of X and $\widehat{X}$. Furthermore, $d_{\text{inv}}(X, \widehat{X}) = 0$ implies that X and $\widehat{X}$ are equal up to a rigid transformation. With this metric, our main result establishes an upper bound on the resulting error. The proof of this theorem is provided in Section VI. We define

$$R_{\text{MDS}} \equiv 32 \left(\frac{24 \log n}{p_0(n-2)} \right)^{\frac{1}{d}} . \tag{5}$$

Theorem 5.1 (connectivity-based model): Assume n nodes are distributed uniformly at random in the $[0,1]^d$ hypercube, for a bounded dimension $d \in \{2, 3\}$. For a positive radio range R with a minimum detection probability p_0, we are given the connectivity information of the nodes according to the range-free model with probabilistic detection. Then, with a probability larger than $1 - 1/n^4$, the distance between the estimate $\widehat{X}$ produced by MDS-MAP and the correct position matrix X is bounded by

$$d_{\text{inv}}(X, \widehat{X}) \leq \frac{R_{\text{MDS}}}{R} + 20R , \tag{6}$$

for $R > (1/p_0)^{1/d} R_{\text{MDS}}$, where $d_{\text{inv}}(\cdot)$ is defined in (4) and R_{MDS} in (5).
The proof is provided in Section VI. The following corollary trivially follows, as for each $(i,j) \in E$, we have $d_{i,j} \leq R$.

Corollary 5.2 (range-based model): Under the hypotheses of Theorem 5.1 and in the case of range-based model, with high probability

$$d_{\text{inv}}(X, \widehat{X}) \leq \frac{R_{\text{MDS}}}{R} + 20R .$$

As described in the previous section, we are interested in the regime where $R = C(\log n/n)^{1/d}$ for some constant C. Given a small positive constant δ, this implies that MDS-MAP is guaranteed to produce estimated positions that satisfy $d_{\text{inv}}(X, \widehat{X}) \leq \delta$ with a large enough constant C and a large enough n. When p_0 is fixed and $R = C(\log n/n)^{1/d}$ for some positive parameter C, the error bound in (6) becomes $d_{\text{inv}}(X, \widehat{X}) \leq \frac{C_1}{C p_0^{1/d}} + C_2 C \left(\frac{\log n}{n}\right)^{1/d}$, for some numerical constants C_1 and C_2. The first term is inversely proportional to C and $p_0^{1/d}$ and is independent of n, whereas the second term is linearly dependent on C and vanishes as n grows large. This is illustrated in Figure 5, which shows numerical simulations with n sensors randomly distributed in the 2-dimensional unit square, and assuming (1). Notice that the resulting error decreases with p_0 and is independent of β.

Even though the upper bounds for both range-free and range-based models have the same form, their behaviours is different as R grows. In the range-free case, up to some point, the performance of MDS-MAP improves as R increases. This

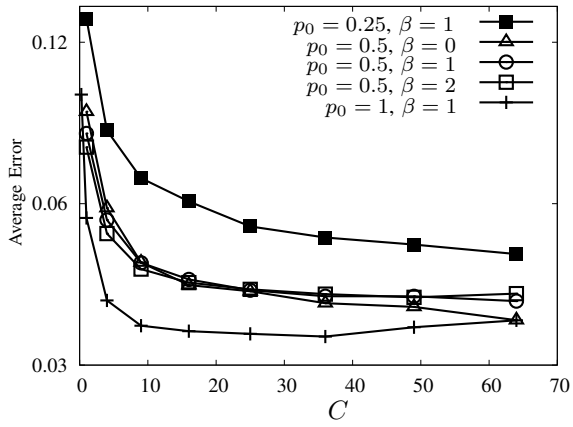

Fig. 5. Average error for MDS-MAP with $R = C\sqrt{\log n/n}$ under range-free model.

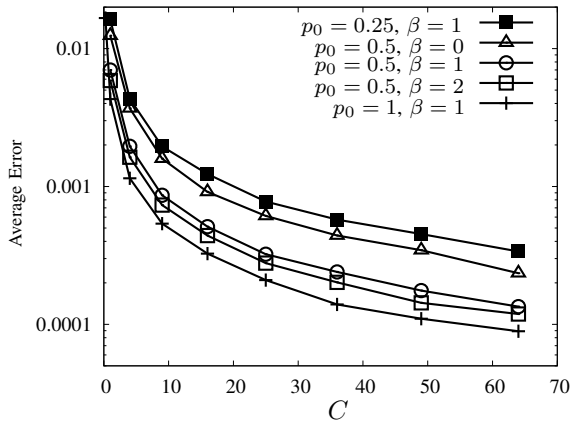

Fig. 6. Average error of MDS-MAP under the range-based model.

is due to the fact that the first and second terms go in opposite directions as a function of R. However, in the range-based case (where we measure the pairwise distances exactly if the pair is within a radio range R), as R increases, we obtain more measurements of the exact Euclidean distances. As a result, once the radio range increases, the resulting error of MDS-MAP decreases and we do not see the contribution of the second term. This phenomenon is illustrated in Figure 6.

Using the above theorem, we can further show that there is a linear transformation $S \in \mathbb{R}^{d\times d}$, such that when applied to the estimations, we get a similar bound in the Frobenius norm of the error in the positions.

Theorem 5.3: Under the hypotheses of Theorem 5.1, with probability larger than $1 - 1/n^4$,

$$\min_{S\in\mathbb{R}^{d\times d}} \frac{1}{\sqrt{n}}\|LX - L\widehat{X}S\| \leq \sqrt{6}\left(\frac{R_{\text{MDS}}}{R} + 20R\right) .$$

A more natural choice of metric might be $\min_{Q\in\mathsf{O}(d)} \frac{1}{\sqrt{n}}\|LX - L\widehat{X}Q\|$. However, there is no known relationship between this quantity and the bound in Theorem 5.1.

Note that although for the sake of simplicity, we focus on $[0,1]^d$ hypercube; our analysis easily generalizes to any bounded convex set and homogeneous Poisson process model with density $\rho = n$. The homogeneous Poisson process model is characterized by the probability that there are exactly k nodes appearing in any region with volume A : $\mathbb{P}(k_A = k) = \frac{(\rho A)^k}{k!}e^{-\rho A}$. Here, k_A is a random variable defined as the number of nodes in a region of volume A.

To simplify calculations, we assumed that d is either 2 or 3. However, the analysis easily applies to general d and only the constant in the bound (6) would change as long as $d = O(1)$.

B. HOP-TERRAIN

Our second result establishes that HOP-TERRAIN [4] achieves an arbitrarily small error for a radio range $R = C(\log n/n)^{1/d}$ with a large enough constant C, when we have only the connectivity information as in the case of the *connectivity-based model*. The same bound holds immediately for the *range-based model*, when we have an approximate measurements for the distances, and the same algorithm can be applied without any modification. To compute better estimates for the actual distances between the unknown nodes and the anchors, the extra information can be readily incorporated into the algorithm. We define

$$R_{\text{HOP}} \equiv 12\left(\frac{24\log n}{p_0(n-2)}\right)^{\frac{1}{d}} . \tag{7}$$

Theorem 5.4: Assume n sensors and m anchors are distributed uniformly at random in the $[0,1]^d$ hypercube for $d \in \{2,3\}$. For a given radio range $R > (1/p_0)^{1/d}R_{\text{HOP}}$, with a minimum detection probability p_0, and the number of anchors $m = \Omega(\log n)$, the following is true with probability at least $1-1/n^4$. For all unknown nodes $i \in V_u$, the Euclidean distance between the estimate $\hat{x}_i$ given by HOP-TERRAIN and the correct position x_i is bounded by

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + 24R . \tag{8}$$

The proof is provided in Section VI. As described in the previous section, we are interested in the regime where $R = C(\log n/n)^{1/d}$ for some constant C. Given a small positive constant δ, this implies that HOP-TERRAIN is guaranteed to produce estimated positions that satisfy $\|x_i - \hat{x}_i\| \leq \delta$ for all i with a large enough constant p_0 and large enough n.

When the number of anchors is bounded and the positions of the anchors are chosen randomly, it is possible that, in the triangulation step, we get an ill-conditioned matrix A^TA, resulting in an large estimation error. This happens, for instance, if three anchors fall close to a line. However, as mentioned in the introduction, it is reasonable to assume that, for the anchors, the system designer has some control over where they are placed. In that case, the next theorem shows that when the positions of anchors are properly chosen, only $d+1$ anchors suffice to get a similar bound. Note that this is the minimum number of anchors necessary for triangulation. For simplicity we assume that one anchor is placed at the origin and d anchors are placed at positions corresponding to d-dimensional unit vectors. The position of the $d+1$ anchors are $\{[0,\ldots,0],[1,0,\ldots,0],\ldots\}$. (see figure 7)

Theorem 5.5: Under the hypotheses of Theorem 5.4, assume that there are $d+1$ anchors, one of which is placed at the origin, and the position vectors of the d remaining anchors

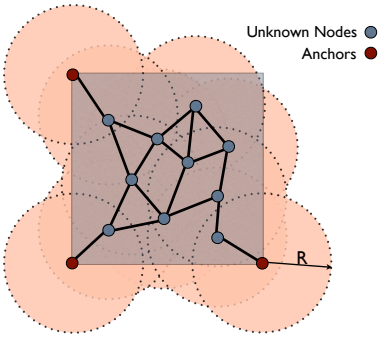

Fig. 7. Three anchors in fixed positions ($[0,0],[1,0],[0,1]$) with $d=2$.

are the d-dimensional unit vectors. Then, the following is true with probability at least $1-1/n^4$. For all $i \in V_u$, HOP-TERRAIN achieves

$$\|x_i - \hat{x}_i\| \leq 2\frac{R_{\text{HOP}}}{R} + 48R\,. \tag{9}$$

The proof is provided in Section VI. There is nothing particular about the position of the anchors in unit vectors. Any $d+1$ anchors in general position will give similar bounds. The only difference is that the constant term in the definition of R_{HOP} changes with the anchor positions.

Consider a case where we are designing a sensor network system for environmental monitoring. We want to decide how many sensors to deploy (dropping them from an air plane) and how much radio power each sensors need to be equipped with, in order to be able to determine the positions of the sensors up to an error of ϵ, for any positive ϵ. Then, our main results in (8) gives us a guideline for choosing appropriate n and R such that the design goal is met. Precisely, choosing $R = \sqrt{R_{\text{HOP}}/24}$ which minimizes the right-hand side of (8) and choosing $n \geq C/(\epsilon^6)$ for some universal constant C achieves the desired design goal. The similar argument is true for a centralized system as well.

There is no known theoretical lower bound for localization under the model considered. One implication of our main results is that we guarantee, with high probability, that for any arbitrarily small constant c, we can achieve error less than c with radio range R which scales as R_{critical}. We cannot hope to achieve a small error bound with $R < R_{\text{critical}}$, since the graph starts to be disconnected. However, as we saw in the above example, the dependence on the desired error c might not be optimal, and there might be better algorithms that can achieve the error with smaller R.

Corollary 5.6 (range-based model): Under the hypotheses of Theorem 5.4 and in the range-based model, with high probability

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + 24R\,.$$

The similar result holds true when sensors are places deterministically, specifically, under the hypothesis of Theorem 5.5, with high probability,

$$\|x_i - \hat{x}_i\| \leq 2\frac{R_{\text{HOP}}}{R} + 48R\,.$$

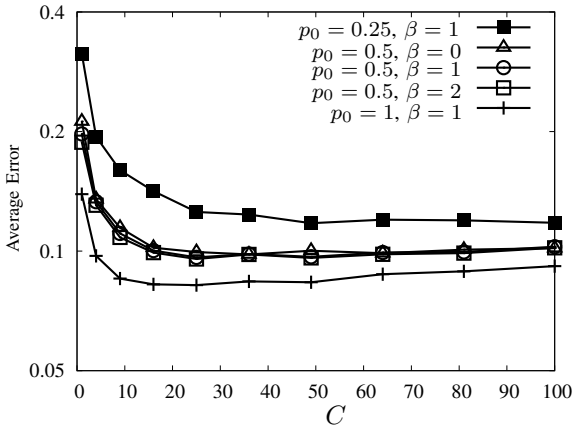

Fig. 8. Average error of HOP-TERRAIN for $R = C\sqrt{\log n/n}$ under connectivity-based model.

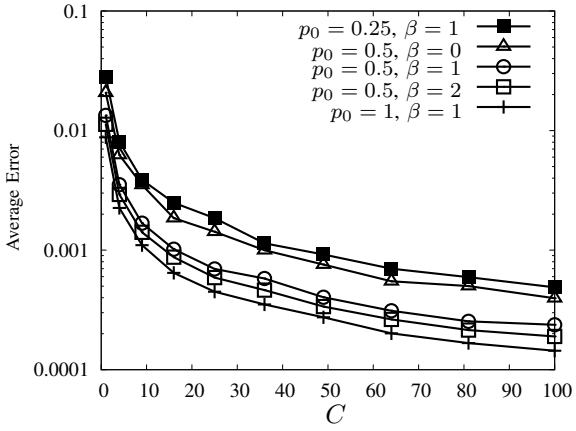

Fig. 9. Average error under range-based model.

As it was the case for MDS-MAP, when $R = C(\log n/n)^{1/d}$ for some positive parameter C, the error bound in (9) is $\|x_i - \hat{x}_i\| \leq \frac{C_1}{Cp_0^{1/d}} + C_2 C\left(\frac{\log n}{n}\right)^{1/d}$, for some numerical constants C_1 and C_2. The first term is inversely proportional to C and $p_0^{1/d}$ and is independent of n, whereas the second term is linearly dependent in C and vanishes as n grows large. This is illustrated in Figure 8, which shows numerical simulations with $n = 5,000$ sensors in the unit square. We compare $\{(1/n)\sum_{i=1}^{n}\|x_i - \hat{x}_i\|^2\}^{1/2}$. The Average error under range-based model is shown in Figure 9.

Figure 10 shows a network consisting of $n = 200$ nodes place randomly in the unit circle. The three anchors in fixed positions are displayed by solid blue circles. In this experiment the distance measurements are from the range-based model and the radio range is $\sqrt{0.8\log n/n}$. In the final estimated positions using HOP-TERRAIN, the circles represent the correct positions, and the solid lines represent the differences between the estimates and the correct positions. The average error in this example is 0.075.

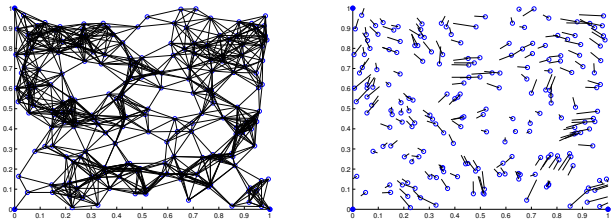

Fig. 10. 200 randomly placed nodes with 3 anchors in blue (left). Location estimation using HOP-TERRAIN (right).

VI. Proof of the Main Theorems

A. Proof of Theorem 5.1

We start by bounding the distance $d_{\rm inv}(X,\widehat{X})$, as defined in Eq. (4). Recall $\|A\|_F$ denotes the Frobenius norm and $\|A\|_2$ denotes the spectral norm. For a rank r matrix A we have $\|A\|_2 \leq \|A\|_F \leq \sqrt{r}\|A\|_2$. Since $L(XX^T - \widehat{X}\widehat{X}^T)L$ has rank at most $2d$, we get

$$\|L(XX^T - \widehat{X}\widehat{X}^T)L\|_F \leq \sqrt{2d}\|L(XX^T - \widehat{X}\widehat{X}^T)L\|_2 \ .$$

To bound the spectral norm, let $M = -(1/2)L\widehat{D}L$. Then,

$$\begin{aligned}\|L(XX^T - \widehat{X}\widehat{X}^T)L\|_2 &\leq \|LXX^TL - M\|_2 + \|M - \widehat{X}\widehat{X}^T\|_2 \\ &\leq (1/2)\|L(-D+\widehat{D})L\|_2 \\ &\quad +(1/2)\|L(-\widehat{D}+D)L\|_2 \\ &\leq \|\widehat{D}-D\|_2 \ , \end{aligned} \tag{10}$$

where in the first inequality we used the triangular inequality and the fact that $\widehat{X} = L\widehat{X}$. In the second inequality we used (2) and the fact that $\|M - \widehat{X}\widehat{X}^T\|_2 = \min_{A:\mathrm{rank}(A)\leq d}\|M - A\|_2$, which follows from the definition of $\widehat{X}$. From the definition of $\widehat{X} = \mathsf{MDS}_d(\widehat{D})$, we know that $\widehat{X}\widehat{X}^T$ is the best rank-d approximation to M. Hence, $\widehat{X}\widehat{X}^T$ minimizes $\|M - A\|_2$ for any rank-d matrix A. Since the rank of $-(1/2)LDL$ is d, this implies $\|M - \widehat{X}\widehat{X}^T\|_2 \leq \|M + (1/2)LDL\|_2$. The inequality (10) follows trivially from the observation that $\|L\|_2 = 1$.

Next, to bound $\|\widehat{D} - D\|_2$, we use the following key result. The main idea is that, the number of hops scaled by the radio range R provides a good estimate of the correct distance. Let $\widetilde{R} \equiv 2(24\log n/(p_0(n-2)))^{1/d}$.

Lemma 6.1: (Bound on the distance estimation) Under the hypotheses of Theorem 5.1, with probability larger than $1 - 1/n^4$, for any pair of nodes $i \in V$ and $j \in V$, the number of hops between nodes i and j is bounded by

$$h_{i,j} \quad \leq \quad \Big(1 + \frac{\widetilde{R}}{R}\Big)\frac{d_{i,j}}{R} + 2 \ ,$$

for $R > \max\{7\widetilde{R}, (1/p_0)^{1/d}\widetilde{R}\}$.

The proof of this lemma is provided in Section VI-G. The distance estimate from the first step of MDS-MAP is $\hat{d}_{i,j} = Rh_{i,j}$. The following corollary gives a bound on the error.

Corollary 6.2: Under the hypotheses of Lemma 6.1,

$$\hat{d}^2_{i,j} - d^2_{i,j} \leq \frac{30\widetilde{R}}{14R}d^2_{i,j} + 8R \ .$$

Proof: From Lemma 6.1, we know that

$$(R\,h_{i,j})^2 - d^2_{i,j} \leq \frac{2\widetilde{R}}{R}\Big(1 + \frac{\widetilde{R}}{2R}\Big)d^2_{i,j} + 2R\Big(1 + \frac{\widetilde{R}}{R}\Big)d_{i,j} + 4R^2 \ .$$

The corollary follows from the assumption that $7\widetilde{R} < R \leq 1$ and $d \leq 3$. ∎

Define an error matrix $Z = \widehat{D} - D$. Then by Corollary 6.2, Z is element-wise bounded by $0 \leq Z_{ij} \leq (30\widetilde{R}/(14R))D_{ij} + 8R$. We can bound the spectral norm of Z as follows. Let u and v be the left and right singular vectors of the non-negative matrix Z, respectively. Then by Perron-Frobenius theorem, u and v are also non-negative. It follows that

$$\begin{aligned}\|\widehat{D} - D\|_2 &= u^TZv \\ &\leq (30\widetilde{R}/(14R))u^TDv + (\mathbb{1}^Tu)(\mathbb{1}^Tv)8R \\ &\leq (30\widetilde{R}/(14R))\|D\|_2 + 8Rn \\ &\leq (30\widetilde{R}/(14R))dn + 8Rn \ . \end{aligned} \tag{11}$$

The first inequality follows from the element-wise bound on Z and the non-negativity of u and v, and the second inequality follows form the definition of the spectral norm and the Cauchy-Schwarz inequality. In the last inequality, we used $\|D\|_2 \leq dn$, which follows from the fact that D is non-negative and element-wise bounded by d. Typically we are interested in the regime where $R = o(1)$, and by assumption we know that $R \geq \widetilde{R}$ and $d \leq 3$. Therefore, the first term in (11) dominates the error. Substituting this bound on $\|\widehat{D} - D\|_2$ in (10) proves the theorem.

B. Proof of Theorem 5.3

Using SVD we can write LX as $U_{n\times d}\Sigma_{d\times d}V^T_{d\times d}$ where $U^TU = \mathbb{I}_{d\times d}$, $V^TV = VV^T = \mathbb{I}_{d\times d}$ and Σ is a diagonal matrix. We also denote the inner product of matrices by $\langle A, B\rangle \equiv \sum_{i,j}A_{i,j}B_{i,j}$. It is easy to show that $\langle A, B\rangle = \mathrm{Tr}(A^TB) \leq \|A\|_F\|B\|_F$. Now, for $S = \widehat{X}^TLU\Sigma^{-1}V^T$, then

$$\|LX - L\widehat{X}S\|_F = \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1}\langle B, LX - L\widehat{X}S\rangle \tag{12}$$

The above equation can be further written as

$$\begin{aligned}(12) &= \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1}\langle B, (LXV\Sigma U^T - L\widehat{X}\widehat{X}^TL)U\Sigma^{-1}V^T\rangle \\ &= \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1}\langle BV\Sigma^{-1}U^T, LXX^TL - L\widehat{X}\widehat{X}^TL\rangle \\ &= \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1}\|BV\Sigma^{-1}U^T\|_F\|LXX^TL - L\widehat{X}\widehat{X}^TL\|_F. \end{aligned}$$

Using the fact $\|A\|_F = \mathrm{Tr}(A^TA)$ and the cyclic property of the trace, i.e., $\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA)$, we obtain

$$\|BV\Sigma^{-1}U^T\|^2_F = \mathrm{Tr}(BV\Sigma^{-2}V^TB^T) \leq \sigma^{-2}_{\min}\|B\|^2_F,$$

where $\sigma_{\min}$ is the smallest singular value of LX. It remains to show that $\sigma_{\min} \geq \sqrt{n/6}$ holds with high probability when nodes are placed uniformly at random. To this end we need to consider two facts. First, the singular values (and in particular the smallest singular value) are Lipschitz functions of the entries. Second, we have $E(LXX^TL) = (n/12)\mathbb{I}_{d\times d}$. By using concentration of measure for Lipschitz functions on bounded independent random variables, the result follows.

C. Proof of Theorem 5.4

In this section we provide the proofs of the theorems 5.4, and proofs of the technical lemmas are provided in the following sections . From Eq. (3), we get

$$\begin{aligned}\|x_i - \hat{x}_i\| &= \|(A^TA)^{-1}A^Tb_0^{(i)} - (A^TA)^{-1}A^Tb^{(i)}\| \\ &\leq \|(A^TA)^{-1}A^T\|_2\|b_0^{(i)} - b^{(i)}\|\,, \end{aligned} \tag{13}$$

First, to bound $\|b_0^{(i)} - b^{(i)}\|$, we use Corollary 6.2. Since $d_{i,j}^2 \leq d$ for all i and j, we have

$$\begin{aligned}\|b_0^{(i)} - b^{(i)}\| &= \Big(\sum_{k=1}^{m-1}\big(d_{i,k+1}^2 - d_{i,k}^2 - \hat{d}_{i,k+1}^2 + \hat{d}_{i,k}^2\big)^2\Big)^{1/2} \\ &\leq 2\sqrt{m-1}\left(\frac{30\widetilde{R}}{14R}d + 8R\right)\,, \end{aligned} \tag{14}$$

Next, to bound $\|(A^TA)^{-1}A^T\|_2$, we use the following lemma.

Lemma 6.3: Under the hypothesis of Theorem 5.4, the following is true. Assuming random anchor model in which $m = \Omega(\log n)$ anchors are chosen uniformly at random among n sensors. Then we have $\|(A^TA)^{-1}A^T\|_2 \leq \sqrt{\frac{3}{m-1}}$, with high probability.

By assumption we know that $R \geq \widetilde{R}$ and $d \leq 3$. By combining (13), (14) and Lemma 6.3 proves Theorems 5.4.

D. Proof of Theorem 5.5

Similarly to the proof of Theorem 5.4, for an unknown node i, and the estimate $\hat{x}_i$ we have

$$\|x_i - \hat{x}_i\| \leq \|(A^TA)^{-1}A^T\|_2\|b_0^{(i)} - b^{(i)}\|\,,$$

We have already bounded the expression $\|b_0^{(i)} - b^{(i)}\|$ in (14). To bound $\|(A^TA)^{-1}A^T\|_2$, we use the following lemma.

Lemma 6.4: Under the hypothesis of Theorem 5.5, the following are true. We assume a deterministic anchor model, where $m = d+1$ anchors are placed on the positions $x_1 = [1,0,\ldots,0], \ldots, x_m = [0,0,\ldots,0]$. Then, $\|(A^TA)^{-1}A^T\|_2 \leq \frac{d}{2}$, with high probability.

This finishes the proof of Theorems 5.5.

E. Proof of Lemmas 6.3 (Random Model)

In order to upper bound $\|(A^TA)^{-1}A\|_2$ we need to lower bound the smallest singular value of A. Let the symmetric matrix B be defined as A^TA. The diagonal entries of B are

$$b_{i,i} = 4\sum_{k=1}^{m-1}(x_{k,i} - x_{k+1,i})^2, \tag{15}$$

for $1 \leq i \leq d$ and the off-diagonal entries as

$$b_{i,j} = 4\sum_{k=1}^{m-1}(x_{k,i} - x_{k+1,i})(x_{k,j} - x_{k+1,j}), \tag{16}$$

for $1 \leq i \neq j \leq d$ where $x_{k,i}$ is the i-th element of vector x_k. In the following lemmas, we show that with high probability, as m increases, the diagonal entries of B will all be of the order of m, i.e., $b_{i,i} = \Theta(m)$, and the off-diagonal entries will be bounded from above by $m^{\frac{1}{2}+\epsilon}$, i.e., $b_{i,j} = o(m)$.

Lemma 6.5: For any $\epsilon > 0$ the diagonal entries of B are bounded as follows.

$$\mathbb{P}\left(|b_{i,i} - 2(m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \leq 4e^{-m^{2\epsilon}}.$$

We use Hoeffding's inequality. To this end, we need to divide the sum in (15) into sums of even and odd terms as follows:

$$b_{i,i} = b_e^i + b_o^i,$$

where

$$b_e^i = 4\sum_{k\in\text{even}}(x_{k,i} - x_{k+1,i})^2, \tag{17}$$

$$b_o^i = 4\sum_{k\in\text{odd}}(x_{k,i} - x_{k+1,i})^2. \tag{18}$$

This separation ensures that the random variables in summations (17) and (18) are independent. Let the random variable z_k^i denote the term $4(x_{k,i} - x_{k+1,i})^2$ in (17). Since $z_k^i \in [0,4]$ and all the terms in b_e^i are independent of each other, we can use Hoeffding's Inequality to upper bound the probability of the deviation of b_e^i from its expected value:

$$\mathbb{P}\left(|b_e^i - (m-1)/3| > 2m^{\frac{1}{2}+\epsilon}\right) \leq 2e^{-m^{2\epsilon}}, \tag{19}$$

for any fixed $\epsilon > 0$. The same bound holds for b_o. Namely,

$$\mathbb{P}\left(|b_o^i - (m-1)/3| > 2m^{\frac{1}{2}+\epsilon}\right) \leq 2e^{-m^{2\epsilon}}. \tag{20}$$

Hence,

$$\begin{aligned}&\mathbb{P}\left(|b_{i,i} - 2(m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \\ &\quad\leq \mathbb{P}\left(|b_e - (m-1)/3| + |b_o - (m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right),\end{aligned}$$

where we used triangular inequality. Applying union bound, this is upper bounded by $4e^{-m^{2\epsilon}}$.

Lemma 6.6: For any $\epsilon > 0$ the off-diagonal entries of B are bounded as follows.

$$\mathbb{P}\left(|b_{i,j}| > 16m^{\frac{1}{2}+\epsilon}\right) \leq 4e^{-m^{2\epsilon}}.$$

Using the Gershgorin circle theorem [30] we can find a lower bound on the minimum eigenvalue of B.

$$\lambda_{\min}(B) \geq \min_i(b_{i,i} - R_i), \tag{21}$$

where $R_i = \sum_{j\neq i}|b_{i,j}|$. Now, let $\mathbb{B}_{ii}$ denote the event that $\{b_{i,i} < 2(m-1)/3 - 4m^{\frac{1}{2}+\epsilon}\}$ and $\mathbb{B}_{ij}$ (for $i \neq j$) denote the event that $\{b_{i,j} > 16m^{\frac{1}{2}+\epsilon}\}$. Since the matrix B is symmetric, we have only $d(d+1)/2$ degrees of freedom. Lemma 6.5 and 6.6 provide us with a bound on the probability of each event. Therefore, by using the union bound we get

$$\begin{aligned}\mathbb{P}\left(\bigcup_{i\leq j}\overline{\mathbb{B}_{ij}}\right) &\leq 1 - \sum_{i\leq j}\mathbb{P}(\mathbb{B}_{ij}) \\ &= 1 - 3d^2e^{-m^{2\epsilon}}.\end{aligned}$$

Therefore with probability at least $1 - 3d^2e^{-m^{2\epsilon}}$ we have

$$b_{i,i} - R_i \geq \frac{2(m-1)}{3} - 16d\cdot m^{\frac{1}{2}+\epsilon}, \tag{22}$$

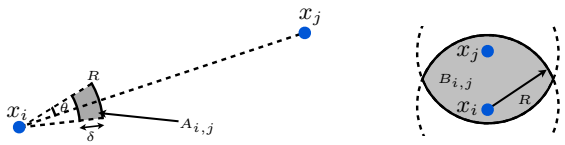

Fig. 11. Two dimensional illustration of a bin $A_{i,j}$ when two nodes are far apart (left), and bin $B_{i,j}$ when two nodes are less than R apart (right).

for all $1 \leq i \leq d$. As m grows, the RHS of (22) can be lower bounded by $(m-1)/3$. By combining (21) and (22) we get

$$\mathbb{P}\left(\lambda_{\min}(B) \geq \frac{(m-1)}{3}\right) \geq 1 - 3d^2 e^{-m^{2\epsilon}}. \tag{23}$$

As a result, from (25) and (23) we have

$$\mathbb{P}\left(\|(A^TA)^{-1}A\|_2 \leq \sqrt{\frac{3}{m-1}}\right) \geq 1 - 3d^2 e^{-m^{2\epsilon}}, \tag{24}$$

which finishes the proof.

F. Proof of Lemmas 6.4 (Deterministic Model)

By using the singular value decomposition of a tall $m-1\times d$ matrix A, we know that it can be written as $A = U\Sigma V^T$ where U is an orthogonal matrix, V is a unitary matrix and Σ is a diagonal matrix. Then, $(A^TA)^{-1}A = U\Sigma^{-1}V^T$. Hence,

$$\|(A^TA)^{-1}A\|_2 = \frac{1}{\sigma_{min}(A)}, \tag{25}$$

where $\sigma_{min}(A)$ is the smallest singular value of A. This means that in order to upper bound $\|(A^TA)^{-1}A\|_2$ we need to lower bound the smallest singular value of A.

By putting the sensors in the mentioned positions the $d \times d$ matrix A will be Toeplitz and have the following form.

$$A = 2\begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

We can easily find the inverse of matrix A.

$$A^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

Note that the maximum singular value of A^{-1} and the minimum singular value of A are related as $\sigma_{min}(A) = (\sigma_{max}(A^{-1}))^{-1}$. To find the maximum singular value of A^{-1}, we need to calculate the maximum eigenvalue of $A^{-1}\left(A^{-1}\right)^T$. Using the Gershgorin circle theorem [30] we can find an upper bound: $\lambda_{\max}\left(A^{-1}\left(A^{-1}\right)^T\right) \leq \frac{d^2}{4}$. Combining this with (25), we get $\|(A^TA)^{-1}A\|_2 \leq \frac{d}{2}$.

G. Proof of the Bound on the Number of Hops

We start by applying a bin-covering technique in a similar way as in [31], [1], [2], but for a more general model where there can be a detection failure. In this section, for simplicity, we assume that the nodes are placed in a 3-dimensional space, but analogous argument is true for $d = 2$ as well.

For each ordered pair of nodes (i, j) we are going to define a corresponding d-dimensional space, which we call a 'bin' (see Figure 11). When two nodes are far apart, i.e. $d_{i,j} > R$, we define a 'bin' as

$$A_{i,j} = \left\{x \in [0,1]^3 \;\middle|\; R-\delta \leq d(x, x_i) \leq R, \measuredangle(x_j - x_i, x - x_i) \leq \theta\right\},$$

where δ and θ are positive parameters which define the size of the bin, and we specify their values later. Here, $\measuredangle(\cdot,\cdot)$ is the angle between two vectors. We say a bin $A_{i,j}$ is occupied if there is a node inside the bin that is detected by node i (i.e., connected to node i in the graph G). We want δ and θ large enough that all bins are occupied (with high probability), but we want them to be small enough that the furthest distance between two nodes in the same bin is small: this distance bounds the maximum error we make in each bin.

Next, when (i, j) is close by, $d_{i,j} \leq R$, we define a bin:

$$B_{i,j} = \left\{x \in [0,1]^3 \;\middle|\; d(x, x_i) \leq R, d(x, x_j) \leq R\right\}.$$

We say a bin $B_{i,j}$ is occupied if there is a node inside the bin that is simultaneously detected by nodes i and j. When n nodes are deployed in $[0, 1]^d$ uniformly at random, we want to ensure that, with high probability, all bins are occupied for appropriate choices of δ and θ as functions of d, R, and p_0, where p_0 is the minimum probability of detection. First when $d_{i,j} > R$,

$$\mathbb{P}\big(A_{i,j} \text{ occupied}\big) = 1 - \prod_{l \neq i,j}\left(1 - \mathbb{P}(\text{node } l \text{ occupies } A_{i,j})\right) \tag{26}$$

The above probability can be lower bounded as follows:

$$\begin{aligned}(26) &\geq 1 - \left(1 - \frac{1}{4}\int_0^\theta \int_{R-\delta}^R 2\pi r^2 \sin(\phi) p_0 \mathrm{d}r\mathrm{d}\phi\right)^{n-2} \\ &= 1 - \left(1 - \frac{1}{2}\pi p_0(1-\cos(\theta))\frac{1}{3}(R^3 - (R-\delta)^3)\right)^{n-2}.\end{aligned}$$

The factor $1/4$ in the first line comes from the fact that, in the worst case the intersection of $A_{i,j}$ and the 3-dimensional cube $[0, 1]^3$ is at most a quarter of the bin $A_{i,j}$, which happens for instance if $x_i = [0, 0, 0]$ and $x_j = [1, 0, 0]$. We choose θ such that $1 - \cos(\theta) = (\delta/R)^2$. Then using the facts that $1 - z \leq \exp(-z)$ and $1 - (1 - z)^3 \geq z$ for $z \in [0, 1]$, we have

$$\mathbb{P}\big(A_{i,j} \text{ is occupied}\big) \geq 1 - \exp\left(-(\pi/6)p_0\delta^3(n-2)\right),$$

which is larger than $1 - 1/n^6$ if we set $\delta = (24 \log n/(p_0(n-2)))^{1/3}$.

Next we consider the case when nodes i and j are at most R apart. Notice that nodes i and j may not be directly connected in the graph. The probability that they are not directly connected is at least p_0, which does not vanish even for large n. But we can show that nodes i and j are at most 2 hops apart with overwhelming probability. Then,

$$\begin{aligned}\mathbb{P}\big(B_{i,j} \text{ is occupied}\big) &= 1 - \prod_{l \neq i,j}\left(1 - \mathbb{P}(\text{node } l \text{ is detected by } i \text{ and } j)\right) \\ &\geq 1 - (1 - V(B_{i,j})p_0^2)^{n-2} \\ &\geq 1 - \exp\left\{-V(B_{i,j})p_0^2(n-2)\right\}, \end{aligned} \tag{27}$$

where $V(B_{i,j})$ is the volume of $B_{i,j}$, and we used the fact that the probability of detection is lower bounded by p_0. $V(B_{i,j})$ is the smallest when nodes i and j are distance R apart and lie on one of the edges of the cube $[0,1]^3$. In a 3-dimensional space, $V(B_{i,j}) \geq (1/4)(5/12)\pi R^3 \geq (1/4)R^3$. Substituting these bounds in (27), we get

$$\mathbb{P}\big(B_{i,j} \text{ is occupied}\big) \geq 1 - \exp\left\{-(1/4)p_0^2 R^3 (n-2)\right\} , \quad (28)$$

which is larger than $1 - 1/n^6$ for $R \geq \left((24 \log n)/((n-2)p_0^2)\right)^{1/3}$.

For each ordered pair (i,j), we are interested in the bin $A_{i,j}$ if $d_{i,j} > R$ and $B_{i,j}$ if $d_{i,j} \leq R$. Using the bounds in (27) and (28) and applying union bound over $n(n-1)$ ordered pairs, all bins are occupied with a probability larger than $1 - 1/n^4$.

Now assuming all bins are occupied, we first show that the number of hops between two nodes i and j is bounded by a function $F(d_{i,j})$ that only depends on the distance between the two nodes. The function $F : \mathbb{R}^+ \to \mathbb{R}^+$ is defined as

$$F(z) = \begin{cases} 2 & \text{if } z \leq R \, , \\ k+2 & \text{if } z \in \mathcal{L}_k \ \text{ for } k \in \{1,2,\ldots\} \, , \end{cases}$$

where $\mathcal{L}_k$ denotes the interval $(k(R-\sqrt{3}\delta) + \sqrt{3}\delta, k(R-\sqrt{3}\delta) + R]$. Our strategy is to use induction to show that for all pairs,

$$h_{i,j} \leq F(d_{i,j}) \, . \quad (29)$$

First, assume nodes i and j are at most R apart. Then, by the assumption that $B_{i,j}$ is occupied there exists a node connected to both i and j. Then the number of hops $h_{i,j}$ is at most two.

Next, assume that (29) is true for all (l,m) with

$$d_{l,m} \leq \sqrt{3}\delta + k(R - \sqrt{3}\delta).$$

For two nodes i and j at distance $d_{i,j} \in \mathcal{L}_k$, consider a line segment $\ell_{i,j}$ in the 3-dimensional space with one end at x_i and the other at x_j. Let $y \in \mathbb{R}^3$ be the point in the line segment $\ell_{i,j}$ that is at distance R from x_i. We want to show that there exists a node that is close to y and is connected to node i. By definition, y is inside the bin $A_{i,j}$. We know that the bin $A_{i,j}$ is occupied by at least one node that is connected to node i. Let us denote one of these nodes by l. Then $d(y, x_l) \leq \sqrt{3}\delta$ because

$$\sup_{z \in A_{i,j}} d(z,y) = \sqrt{\delta^2 + 2R(R-\delta)(1-\cos(\theta))} \leq \sqrt{3}\delta \, ,$$

since we choose $1 - \cos\theta = (\delta/R)^2$.

We use the following triangular inequality: $h_{i,j} \leq h_{i,l} + h_{l,j}$. Since l is connected to i we have $h_{i,l} = 1$. By triangular inequality, we also have $d_{l,j} \leq d(y, x_j) + d(y, x_l)$. It follows from $d(y, x_j) = d_{i,j} - R$ and $d(y, x_l) \leq \sqrt{3}\delta$ that

$$d_{l,j} \leq d_{i,j} - R + \sqrt{3}\delta.$$

Recall that we assumed $d_{i,j} \leq R + k(R - \sqrt{3}\delta)$. Since we assumed that (29) holds for $d_{l,j} \leq \sqrt{3}\delta + k(R - \sqrt{3}\delta)$, we have $h_{i,j} \leq k+2$, for all nodes i and j such that $d_{i,j} \leq R + k(R - \sqrt{3}\delta)$. By induction, this proves that the bound in (29) holds for all pairs (i,j).

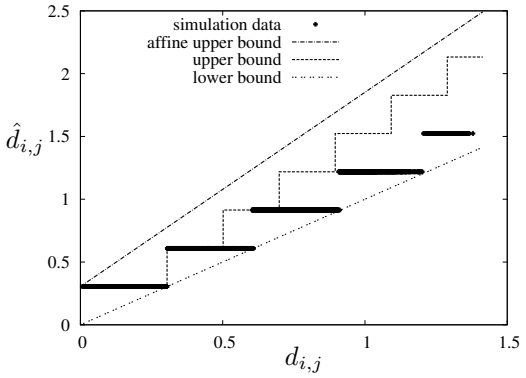

Fig. 12. Comparison of upper and lower bound of shortest paths $\{\hat{d}_{i,j}\}$ with respect to the correct distance $\{d_{i,j}\}$ under connectivity-based model.

We can upper bound $F(z)$ with a simple affine function:

$$\begin{aligned} F(z) &\leq 2 + \frac{1}{R - \sqrt{3}\delta} z \\ &\leq 2 + \Big(1 + \frac{2\delta}{R}\Big)\frac{z}{R} \, , \end{aligned}$$

where the last inequality is true for $R \geq 2\sqrt{3}\delta/(2-\sqrt{3})$. Together with (29) this finishes the proof of the lemma.

Figure 12 illustrates the comparison of the upper bounds $F(d_{i,j})$ and $F_a(d_{i,j})$, and the trivial lower bound $\hat{d}_{i,j} \geq d_{i,j}$ in a simulation with $d = 2$, $n = 6000$ and $R = \sqrt{64 \log n / n}$. The simulation confirms that the shortest paths lie between the analytical upper and lower bounds. Although the gap between the upper and lower bound is seemingly large, in the regime where $R = C\sqrt{\log n / n}$ with a constant C, the vertical gap R vanishes as n grows large and the slope of the affine upper bound can be made arbitrarily small by taking large enough C.

VII. Conclusion

In many applications of wireless sensor networks, it is crucial to determine the location of nodes. For this matter, numerous algorithms have been recently proposed where the efficiency and success of them have been mostly demonstrated by simulations. In this paper, we have investigated the centralized and distributed sensor localization problem from a theoretical point of view and have provided analytical bounds on the performance of such algorithms. More precisely, we analyzed the MDS-MAP and HOP-TERRAIN algorithms and showed that even when only the connectivity information was given and in the presence of detection failure, the resulting error of both algorithms is bounded and decays at a rate inversely proportional to the detection range.

Acknowledgment

We would like to thank Andrea Montanari and Rüdiger Urbanke for stimulating discussions on the subject of this paper.

REFERENCES

[1] S. Oh, A. Karbasi, and A. Montanari, "Sensor network localization from local connectivity : performance analysis for the MDS-MAP algorithm," in *Proc. of the IEEE Inform. Theory Workshop*, January 2010.

[2] A. Karbasi and S. Oh, "Distributed sensor network localization from local connectivity: Performance analysis for the HOP-TERRAIN algorithm," in *ACM SIGMETRICS*, June 2010.

[3] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 201–212, ACM, 2003.

[4] C. Savarese, K. Langendoen, and J. Rabaey, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *USENIX Technical Annual Conference*, (Monterey, CA), pp. 317–328, June 2002.

[5] G. B. Jerey Hightower, "Location Systems for Ubiquitous Computing," 2001.

[6] N. Xu, "A survey of sensor network applications," *IEEE Communications Magazine*, vol. 40, 2002.

[7] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free Positioning in Mobile Ad-Hoc Networks," 2002.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 32–43, ACM, 2000.

[9] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 166–179, ACM, 2001.

[10] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[11] G. MAO, B. FIDAN, and B. ANDERSON, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, pp. 2529–2553, July 2007.

[12] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, pp. 2926–2931, 2001.

[13] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, 2003.

[14] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, 2004.

[15] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, (New York, NY, USA), pp. 46–54, ACM, 2004.

[16] A. Kannan, G. Mao, and B. Vucetic, "Simulated Annealing based Wireless Sensor Network Localization with Flip Ambiguity Mitigation," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, pp. 1022–1026, 2006.

[17] A. Kannan and G. Mao, "Simulated annealing based wireless sensor network localization," *Journal of Computers*, vol. 1, 2006.

[18] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Journal of Telecommunication Systems*, vol. 22, pp. 267–280, 2003.

[19] A. Savvides, H. Park, and M. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mob. Netw. Appl.*, vol. 8, no. 4, pp. 443–451, 2003.

[20] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangant, R. Virrankoski, and A. Savvides, "Distance matrix reconstruction from incomplete distance information for sensor network localization," in *Proceedings of Sensor and Ad-Hoc Communications and Networks Conference (SECON)*, vol. 2, pp. 536–544, Sept. 2006.

[21] M. Fazel, "Matrix rank minimization with applications," *Elec Eng Dept Stanford University*, 2002.

[22] A. Javanmard and A. Montanari, "Localization from Incomplete Noisy Distance Measurements," *arXiv.org*, 2011.

[23] A. Karbasi, S. Oh, R. Parhizkar, and M. Vetterli, "Ultrasound Tomography Calibration Using Structured Matrix Completion," in *The 20th International Congress on Acoustics*, 2010.

[24] E. J. Candès and B. Recht, "Exact Matrix Completion via Convex Optimization," 2008.

[25] R. Keshavan and A. Montanari, "Matrix completion from noisy entries," *The Journal of Machine Learning*, p. 20572078, 2010.

[26] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. Chapman & Hall, 2001.

[27] C. Savarese, J. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *in ICASSP*, pp. 2037–2040, 2001.

[28] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *J. ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[29] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 1, pp. 1106–1110 vol.1, 1998.

[30] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press, 1985.

[31] S. Muthukrishnan and G. Pandurangan, "The bin-covering technique for thresholding random geometric graph properties," in *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 989–998, 2005.

Amin Karbasi (S'10-M'12) earned his PhD in computer and communication sciences from EPFL in September 2012. For his work on the application of graphical models in tomography devices, he received ICASSP 2011 best student paper award. He was also the recipient of ACM/SIGMETRICS 2010 best student paper for his work on distributed sensor localization. His work on network monitoring/tomography was also nominated for the best student paper award in ISIT 2010. His research interests include large-scale statistical inference, graphical models, interactive recommender systems, and information theory.

Sewoong Oh (M'11) is an assistant professor at the department of industrial and enterprise systems engineering at University of Illinois Urbana-Champaign. Before joining UIUC, he worked on statistical inference problems in crowdsourcing at Laboratory for Information and Decision Systems (LIDS), MIT as a postdoctoral associate. He finished his Ph.D. in 2010 at Stanford University working in the area of statistical inference and graphical models with professor Andrea Montanari. He was awarded the Kenneth C. Sevcik outstanding student paper award at the Sigmetrics 2010 for his work in positioning from partial distance measurements.