ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220443023

Connectivity-Based Localization of Large-Scale Sensor Networks with Complex Shape

Article *in* ACM Transactions on Sensor Networks · November 2009

DOI: 10.1145/1614379.1614383 · Source: DBLP

CITATIONS
42

READS
86

3 authors, including:

Jie Gao
Stony Brook University
158 PUBLICATIONS **4,764** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Capacitated Kinetic Clustering in Mobile Networks by Optimal Transportation Theory View project

Dependable Distributed Sensor Systems (D2S2) View project

All content following this page was uploaded by Jie Gao on 30 December 2016.

The user has requested enhancement of the downloaded file.

Connectivity-based Localization of Large Scale Sensor Networks with Complex Shape

Sol Lederer
and
Yue Wang
and
Jie Gao

We study the problem of localizing a large sensor network having a complex shape, possibly with holes. A major challenge with respect to such networks is to figure out the correct network layout, i.e., avoid global flips where a part of the network folds on top of another. Our algorithm first selects landmarks on network boundaries with sufficient density, then constructs the landmark Voronoi diagram and its dual combinatorial Delaunay complex on these landmarks. The key insight is that the combinatorial Delaunay complex is provably *globally rigid* and has a *unique* realization in the plane. Thus an embedding of the landmarks by simply gluing the Delaunay triangles properly recovers the faithful network layout. With the landmarks nicely localized, the rest of the nodes can easily localize themselves by trilateration to nearby landmark nodes. This leads to a practical and accurate localization algorithm for large networks using only network connectivity. Simulations on various network topologies show surprisingly good results. In comparison, previous connectivity-based localization algorithms such as multi-dimensional scaling and rubberband representation generate globally flipped or distorted localization results.

Categories and Subject Descriptors: C.2.1 [**Network Architecture and Design**]: Wireless communication—*C.2.2Computer Systems OrganizationComputer-Communication NetworksNetwork Protocols*; F.2.2 [**Theory of Computation**]: analysis of algorithms and problem complexity—*non-numerical algorithms and problems*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Sensor Networks, Combinatorial Delaunay Complex, Embedding, Localization, Graph Rigidity

1. INTRODUCTION

The physical location of sensor nodes is critical for both network operation and data interpretation. In this paper we focus on anchor-free localization in which none of the nodes know their location and the goal is to recover a relative coordinate system up to global rotation and translation. This is motivated by sensor network applications in remote areas or indoor/underwater environments in which GPS or explicitly placed anchor nodes are not available or too costly. Philosophically,

Department of Computer Science, Stony Brook University, Stony Brook, NY 11794. Email: {lederer,yuewang,jgao}@cs.sunysb.edu.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 20YY ACM 0000-0000/20YY/0000-0001 $5.00

anchor-free localization addresses a very fundamental problem: can we recover the network geometry, simply from the network connectivity information? That is, with local knowledge (knowing which nodes are nearby), can we reconstruct the global picture?

As sensor networks scale in size, retrieving the locations of the nodes becomes even more challenging. The difficulty comes from the network scale, error accumulation, and the increase to both the communication and computation load. Moreover, large deployments of sensor nodes are more likely to have irregular shapes as obstacles and terrain variations inevitably come in to the picture. Our emphasis in this paper is to localize a large sensor network with a complex shape, by using only the network connectivity.

Incorrect flips vs. graph rigidity. A major challenge in network localization is to figure out the correct global layout and resolve flip ambiguities. To give some intuition, Figure 1 illustrates that with only network connectivity information (or even with measurements of the edge lengths), one is unable to tell the "flip" of triangle $\triangle bcd$ relative to a neighboring triangle $\triangle abc$ locally. Both are valid embeddings.

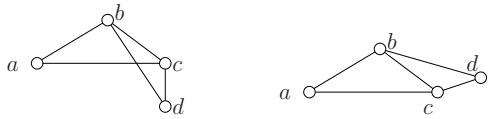


Fig. 1. A connectivity graph with two distinct embedding having the same set of edge lengths.

Figure 2 shows a more severe error, a global flip, that may result from some local flips. The right figure has almost all the nodes correctly localized but has one corner folded over on itself. This is particularly devastating because a node communicating with only its neighbors cannot realize this global error. Indeed, it has been observed that localization algorithms by local optimization may get stuck at one configuration far from the ground truth (see Figure 2 in [Moore et al. 2004]).

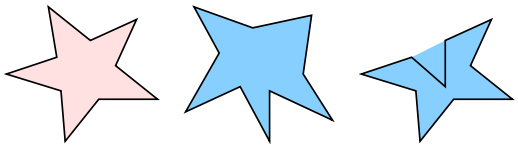


Fig. 2. Left to right: the ground truth; one possible embedding; a more devastating embedding with a global flip.

It thus represents a major difficulty to resolve flip ambiguities in anchor-free localization. When we know the edge lengths, localization is closely related with graph rigidity [Graver et al. 1993] in 2D. A graph is *rigid* if one cannot continuously deform the graph embedding in the plane without changing the edge lengths. A graph is *globally rigid* if there is a unique realization in the plane. Rigidity without global rigidity may yield flip ambiguities. For example, Figure 1 is rigid but not globally rigid. Thus in anchor-free localization, global rigidity is the desirable property.

A number of localization algorithms deal with the problem of rigidity by exploring the graph structure [Eren et al. 2004; Goldenberg et al. 2005; Goldenberg et al.

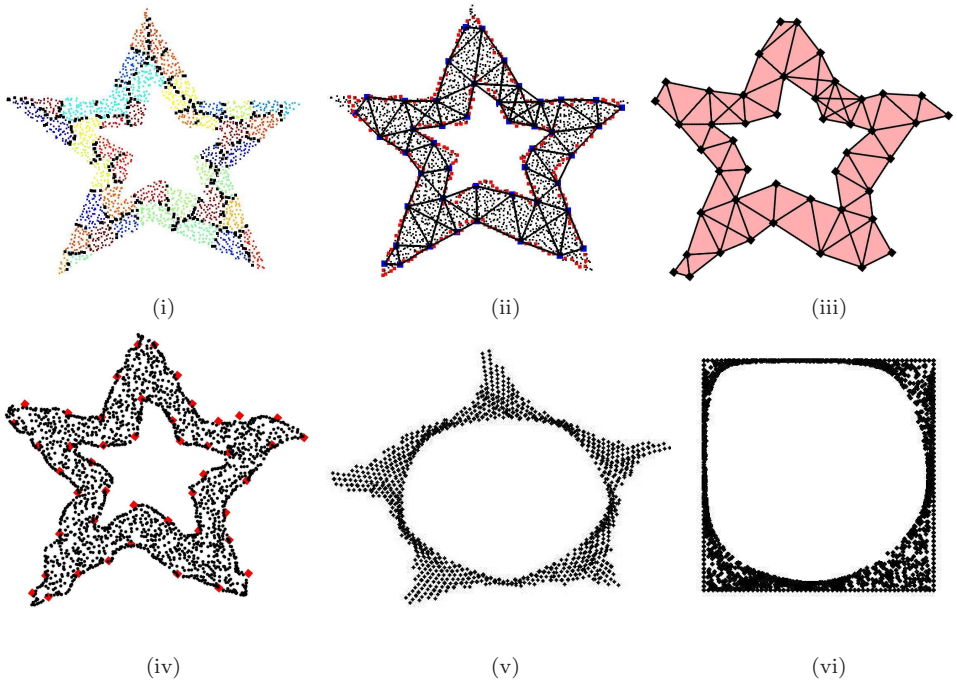


Fig. 3. Anchor-free localization from network connectivity, on a double star shape. The number of nodes is 2171. The connectivity follows a unit disk graph model with average node degree 10. (i) The Voronoi cells of the landmarks (black nodes are on the Voronoi edges); (ii) The Delaunay edges extracted from the Voronoi cells of the landmarks; (iii) Our embedding result of the extracted Delaunay complex; (iv) Our localization result of the entire network. (v) Embedding result by multi-dimensional scaling. (vi) Embedding result by the rubberband representation with the outer boundary fixed along a square.

2006; Moore et al. 2004]. These algorithms either require that the network is dense enough to guarantee the network is a tri-lateration graph[1] (such that iterative tri-lateration method resolves the ambiguity of flips—an even stronger notion than global rigidity) [Eren et al. 2004; Goldenberg et al. 2005; Moore et al. 2004]; or, when the network is sparse, record all possible configurations and prune incompatible ones whenever possible, which, in the worst case, can result in an exponential space requirement [Goldenberg et al. 2006]. All these algorithms require that neighbors are able to estimate their inter-distances, and they do not work with network connectivity alone. Estimating the inter-distances from received signal strength can be quite noisy in a complex environment, and accurate distance estimation requires special ranging hardware.

An approach on anchor-free localization with only network connectivity is to use global optimization such as multi-dimensional scaling (MDS) [Shang et al. 2003]. MDS takes an inter-distance matrix on n nodes and extracts the node location in

[1] A tri-lateration graph G in dimension d is one with an ordering of the vertices $1, \ldots, d+1, d+2, \ldots, n$ such that the complete graph on the initial $d+1$ vertices is in G and from every vertex $j > d+1$, there are at least $d+1$ edges to vertices earlier in the sequence. Tri-lateration graph is globally rigid.

$\mathbb{R}^n$. For 2D embedding, the locations are taken as the largest 2D linear projection. Figure 3 (v) shows the result of (MDS) on figure 3 (vi). Intuitively, MDS tries to stretch the network out in every direction. For a well-connected dense network it gives an effective localization result. But it does not have any notion of rigidity and may produce results with global flips. See more examples in Figure 10.

Discovery of global topology. Aside from localization algorithms, recently there is a growing interest in the study of global topology of a sensor field, and its applications in point-to-point routing and information discovery. The focus is to identify high-order topological features (such as holes) from network connectivity [Funke 2005; Funke and Klein 2006; Fekete et al. 2004; Fekete et al. 2005; Kröller et al. 2006; Wang et al. 2006] and construct virtual coordinate systems with which one can route around holes [Fang et al. 2005; Bruck et al. 2005; Fang et al. 2006; Funke and Milosavljević 2007b; 2007a]. These virtual coordinates are by no means close to the real node coordinates — they are not meant to be close. But one may ask the following question: can the identification of the network geometric features (network boundaries, holes, etc.) help in recovering the true node locations? In other words, with the understanding of the network global topology such as where the holes are, does it allow us to infer some information on graph rigidity that can be used to prevent global flips?

One piece of work that uses network boundaries to generate topologically faithful (i.e., no global folding) embeddings is to use the rubberband embedding, by Rao *et al.* in [Rao et al. 2003] and by Funke and Milosavljevic in [Funke and Milosavljević 2007b]. The idea is to fix the network outer boundary on a rectangle and then each internal node iteratively takes the center of gravity of its neighbors' locations as its own location. The rubberband relaxation converges to what is called the rubberband representation [Tutte 1963]. With the identification of the network outer boundary, this method does give a layout without incorrect folds, but unfortunately induces large distortion as holes are typically embedded much larger than they are. An example is shown in Figure 3 (vi). In the literature [Rao et al. 2003; Funke and Milosavljević 2007b] the rubberband representation is mainly used in assigning virtual coordinates to the nodes for geographical routing purposes and is not used to recover the true node location.

Our contribution. The key idea in this paper is to derive a globally rigid substructure from the extraction of high-order topological features of a sensor field, that recovers the global network layout and provide a basis for a localization algorithm.

We assume the sensor nodes are embedded in a geometric region or on a terrain, possibly with holes. The nodes nearby can directly communicate with each other but far away nodes cannot[2]. We do not use anything beyond the network connectivity information and do not assume neighbors can measure their inter-distances, although such information can be easily incorporated to further improve the localization accuracy.

Briefly, the algorithm can be explained as follows (see Figure 3): Suppose the network boundaries (both the outer boundary and inner hole boundaries) have been

[2]Specifically, in our simulations we have adopted unit disk graph model, quasi-unit disk graph model and probabilistic connectivity model.

discovered (say with any of the algorithms in [Funke 2005; Funke and Klein 2006; Fekete et al. 2004; Fekete et al. 2005; Kröller et al. 2006; Wang et al. 2006]). We take samples on the network boundaries and call them *landmarks*. Each node in the network records the closest landmark in terms of network hop distance. The network is then partitioned into *Voronoi cells*, each of which consists of one landmark and all the nodes closest to it (Figure 3(i)). The Delaunay graph (Figure 3(ii)) as the dual of the Voronoi diagram, has two landmarks connected by a Delaunay edge if their corresponding Voronoi cells are adjacent (or share some common nodes).

Now, here is the key insight: given two *Delaunay triangles* sharing a common edge, there is only *one* way to embed them. Thus there is no flip ambiguity! This is because the Delaunay triangles are induced from the underlying Voronoi partitioning so intuitively we can think them as 'solid' triangles, which, when embedded, must keep their interiors disjoint (the case in Figure 1 left cannot happen). In this paper we make this intuition rigorous. We prove in the case of a continuous geometric domain that when the landmarks are sufficiently dense (with respect to the local geometric complexity), the induced Delaunay graph is rigid. Moreover, the Delaunay complex (with high-order simplices such as Delaunay triangles) is *globally rigid*, i.e., there is a unique way to embed these 'solid' Delaunay triangles in the plane.

The identification of the Delaunay triangles and, more importantly, how to embed them relative to each other overcomes a major hurdle toward anchor-free localization. We use an incremental algorithm to glue the triangles one by one. Each Delaunay edge is given a length equal to the minimum hop count between the two landmarks. Since the hop count is only a poor approximation of the Euclidean distance, we use mass-spring relaxation to improve the quality of the embedding and evenly distribute the error (Figure 3 (iii)).

Now with the landmarks localized and the network layout successfully recovered, the landmarks serve as 'anchor' nodes such that each additional node localizes itself by using trilateration with its hop count distances to 3 or more nearby landmarks (in Figure 3 (iv)).

In our algorithm the discovery of the sensor layout, i.e., landmark selection and discovery of the Delaunay edges is done in a distributed way. The discovered Delaunay complex is delivered to the base station where the embedding of the landmarks is produced. This network layout is then disseminated to the remaining nodes to localize themselves.

The outline of the paper is as follows. In Section 2 we prove the rigidity of the Delaunay complex and describe the criterion for landmark selection, in the case of a continuous domain. Readers can also choose to read Section 3 first, in which we explain the algorithm for the discrete network. Simulation results are presented in Section 4.

2. THEORETICAL FOUNDATIONS

In this section we introduce notations and the theoretical foundation of our algorithm ideas, in particular, the density requirement for landmarks to guarantee the global rigidity of the combinatorial Delaunay complex. Some proofs are put in the Appendix.

2.1 Medial axis, local feature size and r-sample

We consider a geometric region $\mathcal{R}$ with obstacles inside. The boundary $\partial\mathcal{R}$ consists of the outer boundary and boundaries of inner holes. For any two points $p, q \in \mathcal{R}$, we denote by $|pq|$ their Euclidean distance and $d(p, q)$ the *geodesic* distance between them inside $\mathcal{R}$, i.e., the length of the shortest path avoiding obstacles. In a discrete network we can use the minimum hop length between two nodes as their distance, whose analog in the continuous case is the geodesic distance. In this paper all the distances are by default measured by the geodesic distances unless specified otherwise. A ball centered at a point p of radius r, denoted by $B_r(p)$, contains all the points within geodesic distance r from p.

Definition 2.1. *The* medial axis *of $\mathcal{R}$ is the closure of the collection of points, with at least two closest points on the boundary $\partial\mathcal{R}$.*

The medial axis of ∂R consists of two components, one part inside $\mathcal{R}$, called the *inner medial axis*, and the other part outside $\mathcal{R}$, called the *outer medial axis*. See Figure 4. In this paper we only care about the inner medial axis.

We remark that the standard definition of medial axis for curves in the plane measures the Euclidean distance of two points. When we change from Euclidean measure to geodesic measure one may wonder how that changes the inner medial axis. Luckily this is not a big issue as it is not difficult to prove that the inner medial axis under the two measures are the same.

Observation 2.2. *The inner medial axis of $\mathcal{R}$ measured in terms of Euclidean distance is the same as that measured in terms of geodesic distance.*

Now we are ready to explain how to measure the local geometric complexity of $\mathcal{R}$, which determines the sampling density. An example is shown in Figure 4.

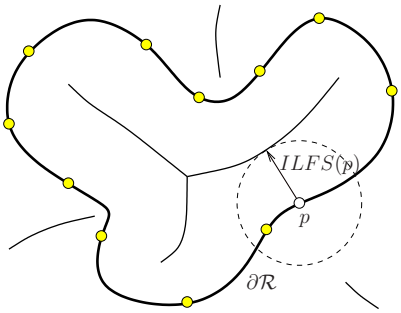


Fig. 4. The region $\mathcal{R}$'s boundary is shown in dark curves. The medial axis and landmarks selected on the boundaries. Point $p \in \partial\mathcal{R}$ has a landmark within distance $ILFS(p)$.

Definition 2.3. *The* inner local feature size *of a point $p \in \partial\mathcal{R}$, denoted as $ILFS(p)$, is the distance from p to the closest point on the inner medial axis. The* local feature size *of a point $p \in \partial\mathcal{R}$, denoted as $LFS(p)$, is the distance from p to the closest point on the medial axis (including both the inner and outer medial axis).*

Definition 2.4. *An r-sample of the boundary $\partial\mathcal{R}$ is a subset of points S on $\partial\mathcal{R}$ such that for any point $p \in \partial\mathcal{R}$, the ball centered at p with radius $r \cdot ILFS(p)$ has at least one sample point inside.*

Landmark density criterion. Our algorithm selects the set of landmarks as an r-sample, with $r < 1$ and selects at least 3 landmarks on each boundary cycle. We will show that these landmarks capture important topological information about the network layout and can be used to reconstruct the network layout.

2.2 Landmark Voronoi diagram and combinatorial Delaunay graph

We take some points in $\mathcal{R}$ and denote them as *landmarks* S. Construct the *landmark Voronoi diagram* $V(S)$ as in [Fang et al. 2005]. Essentially each point in $\mathcal{R}$ identifies the closest landmark in terms of geodesic distance. The *Voronoi cell* of a landmark u, denoted as $V(u)$, includes all the points that have u as a closest landmark:

$$V(u) = \{p \in \mathcal{R} \,|\, d(p,u) \leq d(p,v), \forall v \in S\}.$$

Each Voronoi cell is a connected region in $\mathcal{R}$. The union of Voronoi cells covers the entire region $\mathcal{R}$. A point is said to be on the *Voronoi edge* if it has equal distance to its two closest landmarks. A point is called a *Voronoi vertex* if its distances to three (or more) closest landmarks are the same. A Voronoi edge ends at either a Voronoi vertex or a point on the region boundary $\partial\mathcal{R}$. The *Voronoi graph* is the collection of points on Voronoi edges. The *combinatorial Delaunay graph* $D(S)$ is defined as a graph on S such that two landmarks are connected by an edge if and only if the corresponding Voronoi cells of these two landmarks share some common points. See Figure 5 for some examples. We state some immediate observations

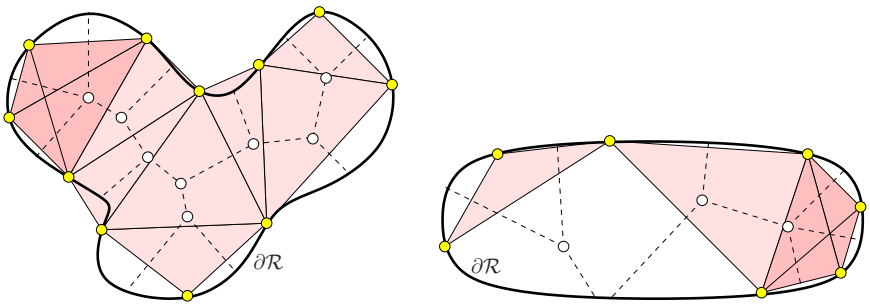


Fig. 5. (i) The Voronoi graph (shown in dashed lines) and the Delaunay graph/complex for a set of landmarks that form an r-sample with $r < 1$. (iii) When the set of landmarks is not an r-sample (with $r < 1$), the combinatorial Delaunay graph may be non-rigid.

about the Voronoi diagram and the corresponding combinatorial Delaunay graph below.

Observation 2.5. *A point on the Voronoi edge of two landmarks u, v* certifies *that there is a Delaunay edge between u, v in $D(S)$. A Voronoi vertex of three landmarks u, v, w* certifies *that there is a triangle between u, v, w in $D(S)$.*

In the case of a degeneracy, four landmarks or more may become cocircular and thus share one Voronoi vertex. See the left top corner in Figure 5 (i). We will capture these high-order features by defining the Delaunay complex in the notion of abstract simplicial complex [Edelsbrunner 2001]. The notion of abstract simplicial complex is defined in a completely combinatorial manner and is described in terms of sets. Formally, a set α is an (abstract) *simplex* with dimension $\dim \alpha = \operatorname{card} \alpha - 1$, i.e., the number of elements in α minus 1. A finite system A of finite sets is an *abstract simplicial complex* if $\alpha \in A$ and $\beta \subseteq \alpha$ implies $\beta \in A$. That is, each set α in A has all its subsets in A as well. In our setting, we construct an abstract simplicial complex from the Voronoi diagram, named the *abstract Delaunay complex*, by taking the *Čěch complex* of the Voronoi cells, defined below.

Definition 2.6. *The* (abstract) Delaunay complex *is the collection of sets*

$$DC(S) = \{\alpha \subseteq S \mid \bigcap_{u \in \alpha} V(u) \neq \emptyset\}.$$

In other words, a set $\alpha \subseteq S$ is a Delaunay simplex if the intersection of the Voronoi cells of landmarks of α is non-empty. The dimension of the Delaunay simplex α is the cardinality of α minus 1.

Thus a landmark vertex is a Delaunay simplex of dimension 0. A Delaunay edge is a simplex of dimension 1. A Delaunay triangle is a simplex of dimension 2 (intuitively, think of the triangle as a 'solid' triangle with its interior filled up). In case of a degeneracy, k landmarks are co-circular and their Voronoi cells have non-empty intersection. This corresponds to a simplex of dimension $k-1$. The rightmost 4 landmarks in Figure 5 (iii) form a dimension-3 simplex (again, intuitively think the simplex as a solid object). We drew the Delaunay complex as shaded regions.

The definition of an abstract simplicial complex is purely combinatorial, i.e., no geometry involved, thus the name of 'abstract' complex. We can talk about an embedding or realization of an abstract simplicial complex (without geometry) in a geometric space as a *simplicial complex* (with geometry). A simplicial complex is geometric and is embedded in a Euclidean space. We give the definitions below. In this paper, we take the abstracted Delaunay complex from the network connectivity graph, and find the geometric realization of the abstract Delaunay complex as a simplicial complex in the plane, thus recovering the global shape of the sensor network.

A finite set of points is *affinely independent* if no affine space of dimension i contains more than $i+1$ of the points, for any i. A *k-simplex* is the convex hull of a collection of $k+1$ affinely independent points S, denoted as $\sigma = \operatorname{conv} S$. The dimension of σ is $\dim \sigma = k$. Figure 6 shows 0, 1, 2, 3-simplex in $\mathbb{R}^3$. The convex

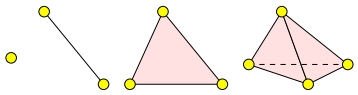


Fig. 6. 0, 1, 2, 3-simplex in $\mathbb{R}^3$.

hull of any subset $T \subseteq S$ is also a simplex. It is a subset of $\operatorname{conv} S$ and called

a *face* of σ. For example, take the convex hull of three points in a 3-simplex, it is a 2-simplex (a triangle). A *simplicial complex* is the collection of faces of a finite number of simplices such that any two of them are either disjoint or meet in a common face. A *geometric realization* of an abstract simplicial complex A is a simplicial complex K together with a bijection φ of the vertex set of A to the vertex set of K, such that $\alpha \in A$ if and only if $\operatorname{conv} \varphi(\alpha) \in K$ [Edelsbrunner 2001]. Of course the ambient space in which the simplicial complex is embedded has to have dimension at least equivalent to the highest dimension of the simplex in A. In our case, when there is degeneracy theoretically we will have to embed in a space with dimension higher than 2. We will discuss how to get around this problem in the next section after the discussion of rigidity. In the rest of the paper, when we say the Delaunay *graph*, we refer to the Delaunay edges and vertices. When we say the Delaunay *complex*, we also include the higher order simplices such as Delaunay triangles and tetrahedrons.

2.3 Global rigidity of combinatorial Delaunay complex

The property of the combinatorial Delaunay graph clearly depends on the selection of landmarks. The goal of this section is to show that the Delaunay graph is rigid when there are at least 3 landmarks on each boundary cycle and they form an r-sample of $\partial\mathcal{R}$ with $r < 1$. In addition, and the Delaunay complex is globally rigid (i.e., it admits a unique 2D realization). An example when the combinatorial Delaunay graph is not rigid due to insufficient sampling is shown in Figure 5 (ii). Now we prepare to prove the rigidity results by first showing that the Voronoi graph (collection of points on Voronoi edges) is connected within $\mathcal{R}$. In this subsection we assume that the landmarks are selected according to the landmark selection criterion mentioned above. The proofs of some of the following Lemmas can be found in the Appendix.

Observation 2.7. *Two Voronoi vertices connected by a Voronoi edge correspond to two Delaunay triangles sharing an edge.*

Lemma 2.8. *For any two adjacent landmarks u, v on the same boundary cycle, there must be a Voronoi vertex inside $\mathcal{R}$ whose closest landmarks include u, v.*

Lemma 2.8 implies that the Delaunay graph has no node with degree 1 – since every node is involved in 2 triangles with its adjacent 2 nodes on the same boundary.

Lemma 2.9. *If there is a continuous curve C that connects two points on the boundary $\partial\mathcal{R}$ such that C does not contain any point on Voronoi edges, then C cuts off a topological 1-disk[3] of $\partial\mathcal{R}$ with at most one landmark inside.*

Corollary 2.10. *The Voronoi graph $V(S)$ is connected.*

Proof. This follows immediately from Lemma 2.8 and Lemma 2.9, although Lemma 2.9 is stronger. Specifically if $V(S)$ is not connected, we are able to find a curve C that cuts $\mathcal{R}$ into two pieces each containing some landmarks and some Voronoi edges, with C not intersecting with the Voronoi graph. □

[3]Intuitively, a topological 1-disk can be continuously deformed into a straight unit length line segment, without any cutting or gluing operations.

Now we are able to show that the combinatorial Delaunay graph is rigid. In other words, given a realization of $D(S)$ in the plane, one cannot deform its shape in the plane without changing the lengths of the edges. To prove this, we use a seminal result about graph rigidity [by G. Laman in 1970], known as the *Laman condition*. It states that generically rigid graphs in 2D can be classified by a purely combinatorial condition. A graph is called a *Laman graph* if it has n vertices, $2n-3$ edges and any subset of k vertices spans at most $2k-3$ edges.

Theorem 2.11 (Laman condition [Laman 1970]). *A graph G with n vertices is generically rigid [4] in 2 dimensions if and only if it contains a Laman graph on n vertices.*

Theorem 2.12. *The combinatorial Delaunay graph $D(S)$ is rigid, under our sampling condition.*

Proof. In this proof we assume without loss of generality that there is no degeneracy, i.e., four or more landmarks are not co-circular. Indeed degeneracy will only put more edges to the combinatorial Delaunay graph, which only helps with graph rigidity.

From the Voronoi graph $V(S)$, we extract a subgraph V' that contains all Voronoi vertices and the Voronoi edges that connect these Voronoi vertices. Some Voronoi edges end at points on the boundary $\partial\mathcal{R}$ and we ignore those. By Corollary 2.10 this graph V' is connected. Now we find a spanning tree T in V' that connects all Voronoi vertices. Take the corresponding subgraph D' of the combinatorial Delaunay graph $D(S)$ such that an edge exists between two landmarks in D' if and only if there is a point in T that certifies it. D' is a subgraph of $D(S)$. Now we argue that D' is a Laman graph.

First the number of landmarks is n. We argue that the number of edges in D' is $2n-3$. Assuming the number of Voronoi vertices is m, T has $m-1$ Voronoi edges. We start from a leaf node on T and sweep along the edges on T. Each time we add one new vertex that is connected to the piece that we have explored through an edge. During the sweep we count the number of landmarks and the number of Delaunay edges that we introduce. To start, we have T' initialized with one Voronoi vertex, thus we have three landmarks and three Delaunay edges. The new Voronoi vertex x we introduce is adjacent to one and only one vertex in T'—if x is adjacent to two vertices in T', then there is a cycle since T' is connected. This will contradict with the fact that T is a tree. Thus in each additional step we will introduce one Voronoi vertex that is connected to T' through one Voronoi edge. This will introduce one new landmark and two new Delaunay edges. When we finish exploring all Voronoi vertices we have a total of $3+(m-1)=m+2=n$ landmarks, and $3+2(m-1)=2n-3$ Delaunay edges between them. Thus D' has n landmarks and $2n-3$ edges.

With the same argument we can show that any subgraph of D' with k landmarks, denoted by S', has at most $2k-3$ edges. This is because a Delaunay edge is certified by a Voronoi edge. Thus we take the Voronoi edges of T whose corresponding

[4] Intuitively, generic rigidity means that almost all (except some degenerate cases) realizations of the graph in the plane are rigid. Generic rigidity is a graph property. However, a generically rigid graph may have some degenerate assignment of edge lengths such that the realization is not rigid.

landmarks all fall inside S'. These Voronoi edges span at most a tree between Voronoi vertices involving only landmarks in S', because they are a subset of a tree T. By the same argument there are at most $2k-3$ edges between landmarks in S'. Thus the graph D' is a Laman graph. By the Laman condition the combinatorial Delaunay graph $D(S)$ is rigid. □

The above theorem shows the rigidity of the combinatorial Delaunay graph, but not the global rigidity yet—there might be several different realizations of the graph in the plane. Indeed for an arbitrary triangulation one may flip one triangle against another adjacent triangle one way or the other to create different embedding. However, this is no longer possible if we embed the *combinatorial Delaunay complex*, induced from the Voronoi diagram $V(S)$. The intuition is that when the triangles are 'solid' and two triangles cannot share interior points there is only one way to embed the Delaunay complex. In the following theorem we show that there can only be a unique way to embed the abstract Delaunay complex. Thus the recovered Delaunay complex does reflect the true layout of the sensor field $\mathcal{R}$.

Recall that we want to find an embedding of the abstract Delaunay complex in 2D. That is, we want to find a mapping φ of the vertices in the plane such that any abstract simplex $\sigma \in DC(S)$ is mapped as a simplex $\operatorname{conv}\varphi(\sigma) \in \mathbb{R}^2$. Notice that in the case of degeneracy there are high-order k-simplices, $k \geq 3$, for which a geometric realization requires embedding into a space of dimension k or higher. However, this is not really a problem if we force the dimension to be 2. Indeed, look at all the edges of a k-simplex, $k \geq 3$, they form a complete graph of $k+1 \geq 4$ vertices. Thus it is a 3-connected graph and redundantly rigid (a graph remains rigid upon removal of any single edge). Existing results in rigidity theory [Hendrickson 1992; Berg and Jordán 2003] show that a graph is globally rigid (uniquely realizable) in 2D under edge lengths constraints if and only if it is tri-connected and is redundantly rigid. Thus all high-order simplices have unique embedding in the plane (up to global translation and rotation). In this paper, we find a geometric realization of the abstract Delaunay complex in the plane. For all the simplices with dimension 2 or smaller, they are mapped to simplices in the plane. For simplices of dimension 3 or higher, the induced *graph* is globally rigid and subject to a unique embedding, as explained above.

Now the Delaunay complex is composed of a set of Delaunay triangles (2-simplices) and high-order simplices (and their sub-simplices, of course). We already know that the high-order simplices are embedded in the plane as globally rigid components. The Delaunay 2-simplices/triangles are embedded as a geometric complex, i.e., the geometric realization of the abstract Delaunay complex. What is left is to show that given two Delaunay triangles $\triangle uvw$ and $\triangle uvp$ sharing an edge, there is only one way to embed them in the plane as required by the definition of simplical complex—that is w and p are on opposite sides of the shared edge uv, as in Figure 7(i). Otherwise, w and p are embedded on the same side of uv. Then either w is inside $\triangle uvp$ (as in Figure 7 (iii)), or p is inside $\triangle uvw$, or two edges intersect at a non-vertex point (as in Figure 7 (ii)). This will violate the properties of a simplicial complex that any two simplices are either disjoint or meet at a common face. If w is inside $\triangle uvp$, then the two simplices, a 0-simplex w and a 2-simplex $\triangle uvp$ intersect at a vertex w which is not a face of $\triangle uvp$. In the other case, if two

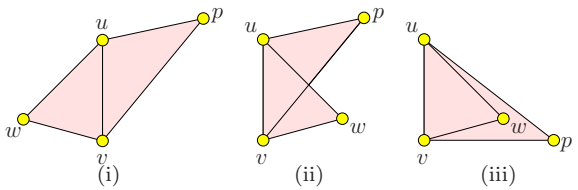


Fig. 7. Two Delaunay triangles $\triangle uvw$ and $\triangle uvp$ sharing an edge. (i) is the only valid valid embedding with the two triangles not sharing any interior points.

edges intersect at a non-vertex point, this intersection is not a face of either edge.

Now we can conclude with the main theoretical result:

Theorem 2.13. *Under our landmark selection criterion, the combinatorial Delaunay complex $DC(S)$ has a unique embedding in the plane up to a global translation and rotation.*

2.4 Topological equivalence

Our sampling condition aims to capture the geometric complexity of the region $\mathcal{R}$. A related question may ask whether the constructed geodesic Delaunay complex is homotopy equivalent[5] to the region $\mathcal{R}$. Homotopy equivalence intuitively says that the number of holes and how they are connected in the Delaunay complex are the same as those in $\mathcal{R}$. Our current sampling condition, unfortunately, can not guarantee the homotopy equivalence. A bad example is shown in Figure 8. To see why this is bad note that, the Voronoi edge of the two landmarks x, y is not simply connected, with two components, one above the small hole in the middle and one below the small hole. Thus the small Delaunay triangle $\triangle xyz$ sticks out of the paper and can not be embedded in the plane. There is no valid geometric realization in the plane without violating the properties of a simplicial complex. The investigation of the sampling condition to guarantee the homotopy equivalence of the geodesic Delaunay complex with the region $\mathcal{R}$ is the topic of a later paper [Gao et al. 2008], in which homotopy feature size and sampling methods to guarantee the topological equivalence are proposed.

With our sampling condition we can still deal with this problem in the following way. As will be shown in the next section, we are able to detect that whether the Voronoi edges adjacent to one Voronoi cell is connected or not. As the following theorem shows, as long as the Voronoi edge/vertex set of any k landmarks is either empty or contractible[6], the homotopy equivalence is established. Thus we can check locally whether the conditions are satisfied.

Theorem 2.14. *If the Voronoi cell/edge/vertex set of any k landmarks is either empty or contractible, the Delaunay complex has the same homotopy type as the region $\mathcal{R}$.*

[5] Two maps f and g from X to Y are homotopic if there exists a continuous map $H : X \times [0, 1] \mapsto Y$ with $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$. Two spaces X and Y have the same homotopy type if there are continuous maps $f : X \mapsto Y$ and $g : Y \mapsto X$ such that $g \circ f$ is homotopic to the identity map of X and $f \circ g$ is homotopic to the identity map of Y. In other words, the maps f and g define a one-to-one correspondence of the topological features such as connected components, cycles, holes, tunnels, etc., and how these features are related.

[6] A set in $\mathbb{R}^d$ which can be reduced to one of its points by a continuous deformation is contractible.

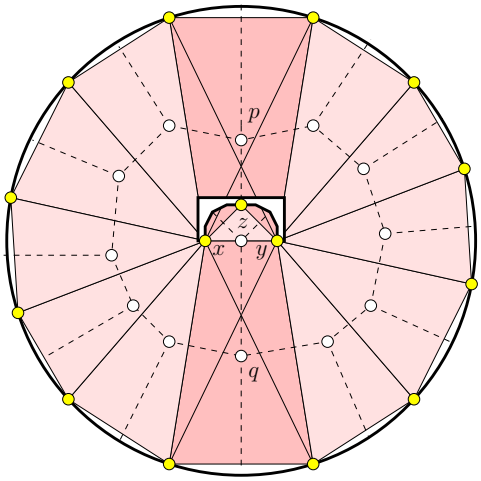


Fig. 8. A nasty example with no valid embedding of the Delaunay complex.

PROOF. As the combinatorial Delaunay complex is the Cěch complex of the Voronoi cells, the theorem follows immediately from the Cěch Theorem [Bott and Tu 1982]. Recall the definition of the Cěch complex. Given a collection of sets $\mathcal{U} = \{V(u), \forall u \in S\}$, the *Cěch complex* is the abstract simplicial complex whose k-simplices correspond to nonempty intersections of $k+1$ distinct elements of $\mathcal{U}$. The Cěch Theorem says that if the sets and all non-empty finite intersections are contractible, then the union $\cup_u V(u)$ has the same homotopy type as the Cěch complex. In our case, the Cěch complex is the Delaunay complex $DS(S)$, the union of the Voronoi cells is $\mathcal{R}$. Thus the claim is true. □

In case of a bad scenario, for our application we can still embed the Delaunay complex in the following way. The embedding would theoretically violate the simplicial complex definition but in practice would be perfectly fine. One thing we notice is that we do know how to embed the triangle $\triangle xyz$ in Figure 8 because the Voronoi vertex of $\triangle xyz$ is connected through a Voronoi edge to the Voronoi vertex q below it. Thus we will embed $\triangle xyz$ so that it is disjoint from the dual simplex of q. But $\triangle xyz$ can and does overlap with the dual simplex of p, since p is not directly connected through a Voronoi edge to the Voronoi vertex of $\triangle xyz$. In other words, we embed the simplices with guidance from the connectivity of the Voronoi vertices that certify them. This is also what we use in the algorithm below.

3. ALGORITHM DESCRIPTION

We assume a large number of sensor nodes scattered in a geometric region. In general nearby nodes can directly talk to each other and far away nodes can not but the algorithm does not strictly enforce a unit disk graph model. The algorithm basically realizes the landmark selection and embedding described in the previous section. Thus we will not re-iterate many things said already and instead focus on the implementation and robustness issues, for the geodesic distance is only poorly approximated by the minimum hop count between two nodes.

We first outline the algorithm and explain each step in detail.

Select landmarks. Nodes on the network boundaries are identified and connected into boundary cycles surrounding inner holes and the outer face by a boundary detection algorithm [Wang et al. 2006]. The inner medial axis is also identified during this process. Along the boundary, landmarks are selected with sufficient density such that for any node p on the boundary, there is a landmark within the inner local feature size $ILFS(p)$ of p, that is, the distance from p to its closest node on the inner medial axis.

Compute landmark Voronoi diagram. The landmarks flood the network and each node records the closest landmark. This generates the Voronoi diagram of the landmarks in a distributed fashion.

Extract the combinatorial Delaunay complex. Nodes on the Voronoi edges/vertices report to their corresponding landmarks. Thus landmarks learn their adjacent Delaunay simplices. Equivalently, this procedure identifies the combinatorial Delaunay complex G. A total of k landmarks are included in a Delaunay simplex if their Voronoi cells share a common node; See Figure 3(i).

Embed the combinatorial Delaunay complex. We apply an incremental algorithm to embed the combinatorial Delaunay complex by gluing these simplices together. We also use mass spring relaxation to improve the embedding result by smoothing out noise in the input.

Network localization. With the embedding of the landmarks we can easily embed the rest of the nodes by trilateration with hop count distances to 3 embedded landmarks.

3.1 Select landmarks

We use a distributed boundary detection algorithm that identifies nodes on both outer and inner boundaries and connects them into boundary cycles [Wang et al. 2006]. With the boundary detected we can identify the medial axis of the sensor field, defined as the set of nodes with at least two closest boundary nodes [Bruck et al. 2005]. The boundary nodes flood inward at roughly the same time [Elson 2003; Ganeriwal et al. 2003]. The flooding messages are suppressed by the hop count to the boundary nodes to reduce message complexity. Specifically, each node records the minimum hop count from the boundary nodes. If a node receives a message containing a hop count no smaller than what it has stored already, the message will be discarded. Otherwise the minimum hop count to the network boundary is updated and the message is further forwarded. Each node learns its closest boundary node. The nodes at which the flooding frontiers collide are nodes on the inner medial axis.

In a discrete network, the medial axis may contain a lot of noises due to the discrete hop count values. For example, a node that is a neighbor of adjacent two boundary nodes is identified to be on the medial axis according to the definition, and is clearly not what we want. There are a number of heuristic algorithms in the past literature to 'clean up' the medial axis of a discrete network [Bruck et al. 2005; Zhu et al. 2007]. The idea is to take the nodes with two or more closest intervals on the network boundary [Zhu et al. 2007]. A node having its closest points on the boundary in a consecutive interval is not identified as the medial axis node.

With the boundary and medial axis identified, we select landmarks from boundary nodes such that for any node p on the boundary, there is a landmark within

distance $ILFS(p)$, where $ILFS(p)$ is the inner local feature size of p defined as the hop count distance from p to its closest node on the inner medial axis. In order to find the local feature size of each node on the boundary, nodes on the medial axis flood the network at roughly the same time with proper message suppression. Each boundary node learns its local feature size as the hop count to its closest node on the medial axis.

Now, landmark selection can be performed by a message traversing along the boundary cycles and select landmarks along the way in a greedy fashion to guarantee the sampling criterion. For each boundary cycle, a node (say the one with minimum ID) marks itself as a landmark and sends a message along the boundary cycle. The message goes as far as possible until for some boundary node p, the message has walked $ILFS(p)$ hops along the boundary from the previously selected landmark. At that point p is marked as a landmark. Keep on going along the boundary cycle until the message comes back to the start node. In this way, landmarks are selected with the desired density. Alternatively, we can let each boundary node p wait for a random period of time and select itself as a landmark. Then p sends a suppression message with TTL as $ILFS(p)$ to adjacent boundary nodes. A boundary node receiving this suppression message will not further select itself as landmarks. Thus landmarks are selected with the required density.

3.2 Compute Voronoi diagram and combinatorial Delaunay complex

The landmark Voronoi diagram is computed in a distributed way as in [Fang et al. 2005]. Essentially all the landmarks flood the network simultaneously and each node records the closest landmark(s). Again a node p will not forward the message if it carries a hop count larger than the closest hop count p has seen. Thus the propagation of messages from a landmark ℓ is confined within ℓ's Voronoi cell. All the nodes with the same closest landmark are naturally classified to be in the same cell of the Voronoi diagram. Nodes with more than one closest landmarks stay on Voronoi edges or vertices.

Unlike the Euclidean case that there is always a point with equal distance to any two or three landmarks, when we adopt the integer hop count measurement as the distance metric, there may not be a point with equal distance to two or three landmarks. Thus we re-define Voronoi vertices in the discrete setting.

Definition 3.1. *An* interior node *is a node p with distance to its closest landmark strictly smaller than its distances to all the other landmarks. A* border node *is a node that is not an interior node.*

Figure 3 (i) is an example of the landmark Voronoi diagram with different Voronoi cells colored differently. Border nodes are colored black. We group these border nodes into Voronoi edges and vertices, i.e., the k-witnesses of $(k-1)$-simplices.

Definition 3.2. *A* k-witness *is a border node which is within 1-hop from interior nodes of k different Voronoi cells. The border nodes that witness the same set of Voronoi cells are grouped into connected* clusters.

One subtle robustness issue, due to the discreteness of sensor nodes, is that there might not be a node that qualifies for the witness defined above (especially for high-order simplices). Thus we propose a *merge* operation: For two clusters A and B

that are both k-witnesses, if there exists a node p in cluster A, or there exists a node q in cluster B, and all nodes in cluster B are neighbors of p or all nodes in cluster A are neighbors of q, then we merge cluster A and B into one cluster that certifies the union of their corresponding landmarks. The benefit of doing so is to generate high order Delaunay simplices even when there are no corresponding witnesses due to the discrete resolution. The above algorithm to identify the abstract Delaunay complex is a heuristic algorithm that uses the intuition from the continuous case. Alternatively we can use the notion of the witness complex [de Silva 2003; Carlsson and de Silva 2004]. This is explored in a later paper [Gao et al. 2008].

The witnesses certify the existence of Delaunay simplices and by definition can be identified locally. A k-witness node w, after it identifies itself, reports to the corresponding landmarks. Such a report contains the IDs of the landmarks involved in this dimension $k-1$ Delaunay simplex, together with the distance vector from the witness node w to each of the k landmarks. Remember that nodes in a Voronoi cell store their minimum hop count distances to their home landmark. Thus, the report just follows the natural shortest path pointer to the landmarks involved (so routing is simple). It can happen that multiple witnesses certify the same Delaunay simplex (say, in the case of a Delaunay edge) and they individually report to the same landmark. These report messages are again suppressed during routing. If a node sees a report about a previously received Delaunay simplex, it will not forward it. Naturally the report from the witness with the smallest hop count to its landmarks will arrive the earliest. With these reports, a landmark learns the combinatorial Delaunay simplices it is involved in, and in addition, an approximate hop count to the other landmarks in those simplices through the distance vectors carried in the reports. In particular, a landmark p estimates the hop count distance to landmark q as the minimum of the sum of distances from the witness node to p and q, over all reports received with q involved. This distance estimation can be directly used to embed the Delaunay simplices. Alternatively, if the minimum hop count distances between neighboring landmarks are desired, one can let the messages initiated by the landmarks travel to the adjacent Voronoi cells. Thus each landmark learns the minimum hop count to all neighboring landmarks.

We remark that in the protocol we aggressively use message suppression to reduce the communication cost. With reasonable synchronization most of the flood messages are pruned and the average number of messages transmitted by each node is within a small constant. We also remark that local synchronization (with possible global clock drifts) is sufficient as message suppression occurs mostly among neighboring landmarks.

3.3 Embed Delaunay complex

Now we are ready to glue the simplices together to embed the landmarks and generate the network layout. Since there is only one way to glue two adjacent simplices (to keep their interiors disjoint, as shown by Theorem 2.13), the embedding is unique. We first embed one simplex S_1 arbitrarily. Then we can embed its neighbor S_2 as follows: Let ℓ_1 and ℓ_2 be the landmarks they share in common. Since S_1 and S_2 are adjacent, such landmarks must exist. For each landmark ℓ_i in S_2 not yet embedded, we compute the 2 points that are with distance $d(\ell_1, \ell_i)$ from ℓ_1 and $d(\ell_2, \ell_i)$ from ℓ_2, where $d(\cdot, \cdot)$ is the hop-count distance between landmarks,

estimated in the previous section. Among the two possible locations we take the one such that the orientation of points $\{\ell_1, \ell_2, \ell_i\}$ is different from the orientation of $\{\ell_1, \ell_2, \ell_r\}$, where ℓ_r is any landmark of S_1, other than ℓ_1 and ℓ_2. Thus ℓ_i and ℓ_r lie on opposite sides of edge $\ell_1\ell_2$.

In some cases one landmark may have two or more neighboring simplices that are already embedded and is thus given multiple coordinate assignments. A natural solution is to take ℓ at the centroid of the different positions. After we have a rough embedding of the entire Delaunay complex, we apply a mass-spring algorithm [Kobourov et al. 2006; Howard et al. ; Kamada and Kawai 1989; Fruchterman and Reingold 1991; Priyantha et al. 2003] to "smooth out" the disfigurements caused by the conflicting node assignments. It is important to recognize however, that mass-spring plays a minor role in our algorithm and its utility is only apparent here because we initially start with topologically correct landmarks positions, i.e., no global flips. Without this initial configuration with good layout a naive mass-spring algorithm can easily gets stuck at local minima, as observed by many [Kobourov et al. 2006; Priyantha et al. 2003].

Briefly, the idea of mass-spring embedding is to think of the landmarks as masses and each edge as a spring, whose length is equal to the estimated hop count distance between two landmark nodes. The springs apply forces on the nodes and make them move until the system stabilizes. The objective is to have the measured distances (based on their current locations) between landmarks match as closely as possible the expected distances (indicated by hop count values). For landmark ℓ_i we let p_i designate its current position, and let $d(i,j)$, $r(i,j)$ be the estimated and measured distance between ℓ_i and ℓ_j, respectively. Each edge creates a force $F = (d(i,j) - r(i,j))/d(i,j)$ along the direction p_ip_j. So the total force on landmark ℓ_i is $F_i = \sum F_{ij}$ for all neighbors ℓ_j. And the total "energy" of the network is $E = \sum (d(i,j) - r(i,j))^2$. We iteratively modify the node positions, based on the forces acting upon them, until the energy of the system ceases to decrease.

We remark that this heuristic embedding algorithm only guarantees that adjacent Delaunay triangles are embedded 'side-by-side'. It does not prevent two chains of triangles from wrapping around and overlapping each other. In fact, given a planar graph with specified edge lengths, it is a NP-hard problem to find a planar embedding [Bateni et al. 2007; Cabello et al. 2007]. Our problem is more difficult as we only have approximate edge lengths. It remains as future work to develop efficient approximation algorithms to embed a planar graph with approximate edge lengths.

In a distributed environment the embedding of the Delaunay simplices can be done incrementally with message passing. Alternatively, the combinatorial Delaunay complex can be collected at a central station where the embedding is performed and disseminated to the remaining nodes. As the number of landmarks is only dependent on the geometric complexity of the sensor field, it is much smaller than the total number of nodes. Thus a centralized collection and dissemination of the landmark positions is manageable.

Recall that after the witnesses report to the relevant landmarks, the landmarks have the information about the Delaunay simplices they are involved in. Thus each landmark can embed its adjacent Delaunay simplices in a local coordinate

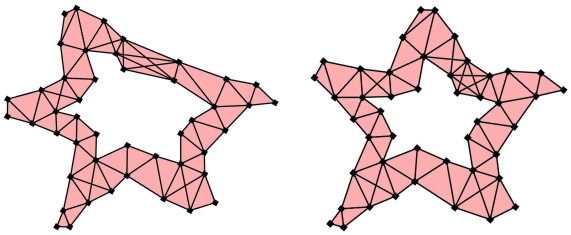


Fig. 9. left: before the mass-spring relaxation algorithm is applied; Right: after mass-spring relaxation.

frame. Then one landmark can initiate a message carrying the partially embedded Delaunay complex to its neighboring landmark. As this message is passed around, more simplices are glued together. Remember there is no ambiguity of how two simplices should be assembled even when the assembly is performed separately at different landmarks. At the end of the message passing mass spring relaxation can be performed to improve the quality.

3.4 Network localization

With the global network layout faithfully recovered, embedding of the rest of non-landmark nodes is easy. Since the locations of the landmarks are known, each non-landmark node just runs a tri-lateration algorithm to find its location (e.g., the atomic trilateration in [Savvides et al. 2001]) by using the hop count estimation to 3 or more landmarks. We also performs a couple rounds of rubberband relaxation to further improve the localization quality for the remaining nodes. An even simpler scheme is to align the boundary nodes along the boundaries of the embedded combinatorial Delaunay complex and perform a rubberband relaxation for the rest of the nodes.

4. SIMULATIONS

We conducted simulations on various network topologies and node densities to evaluate our algorithm and compare with existing solutions.

4.1 Simulation setup and models

In the simulations we use three different models for the network connectivity.

(1) *Unit disk graph model:* two nodes are connected by an edge if and only if the Euclidean distance between them is no greater than 1.
(2) *Quasi-unit disk graph model:* two nodes are connected by an edge if the Euclidean distance between them is no greater than a parameter α, $\alpha < 1$, and are not connected by an edge if the Euclidean distance is larger than 1. If the Euclidean distance d is in the range $(\alpha, 1]$, there may or may not be an edge between them. We include this edge with probability $(1 - d)/(1 - \alpha)$.
(3) *Probabilistic connectivity model:* with unit disk graph model, we additionally remove each edge with probability q.

The nodes are distributed according to a perturbed grid distribution. Each node is perturbed from the grid point with a uniform distribution. That is, for any node

$p(x, y)$ on the grid, we created two random numbers r_x and r_y between 0 and the grid width. Then we use $(x + r_x, y + r_y)$ as the node position. We then control the communication range to vary the average node degree.

To vary the network "shape", We tried different network topologies by including single or multiple holes, convex or concave holes, and some difficult cases such as a U-shape or a Sprial-shape. The network setup and parameters are shown in the caption for each topology.

4.2 Algorithms in comparison

Since most localization algorithms assume node inter-distance measurements and/or anchor nodes, to make a fair comparison we only compare with two algorithms that also use network *connectivity information only*:

Multi-dimensional scaling (MDS). Multidimensional scaling has been used by Shang et al. [Shang et al. 2003] for sensor network localization with connectivity information only. It is also the only anchor-free localization algorithm so far using connectivity information. For n nodes, the input to MDS is the pairwise distance estimation of size $O(n^2)$. If the inter-node Euclidean distances are known exactly, then MDS would precisely determine the coordinates of the points (up to global transformations). In this case, since only rough hop-count distances are known, MDS has trouble capturing a twist within the graph, making a long narrow graph not differentiable from a spiral-shaped graph. In addition, MDS is a centralized algorithm and can not be executed in sensor nodes with limited resources. At the heart of MDS is singular value decomposition (SVD) which has a time complexity of $O(n^3)$. In our simulation we tested MDS in two cases, once on all the nodes and once on the landmarks only. They produce similar layout results. MDS on all nodes is very slow. For some experiments with 5000 nodes the matrix operation involved in MDS requires more than 1GB memory. This computation is only feasible on powerful nodes such as the base station.

Rubberband representation. In rubberband embedding [Funke and Milosavljević 2007b; Rao et al. 2003], first the perimeter nodes are fixed to a square, for instance. Then each non-perimeter node, v, repeatedly updates its coordinates (x_v, y_v) as the average of the locations of its neighbors. The process stabilizes at the rubberband representation. While the rubberband representation is able to avoid global flips if the outer boundary is detected correctly, the shape of the sensor field is wildly distorted. In our experiments the rubberband representation does not give enlightening results on the network layout. Examples are given in Figure 3 (iv) and Figure 11.

4.3 Simulation results

The objective of the following simulations is to evaluate our algorithm and compare with MDS or rubberband representations. In particular, we would like to investigate how does the algorithm performance depend on different factors such as the network shape, the node density, landmark density, and communication models.

4.3.1 *Influence of network shapes.* We applied our algorithm to a number of networks with different layouts, or "shapes". We observed that the performance

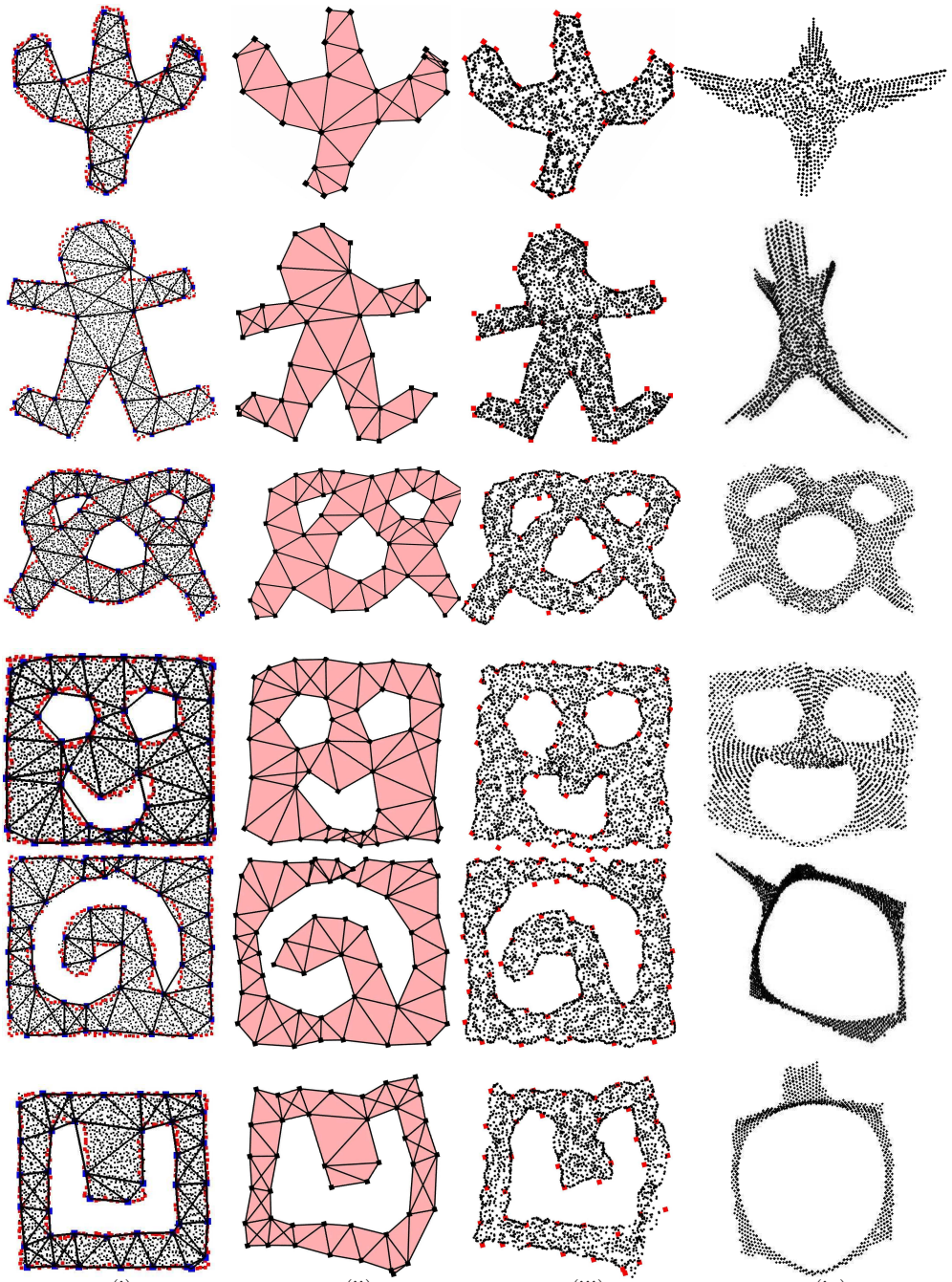
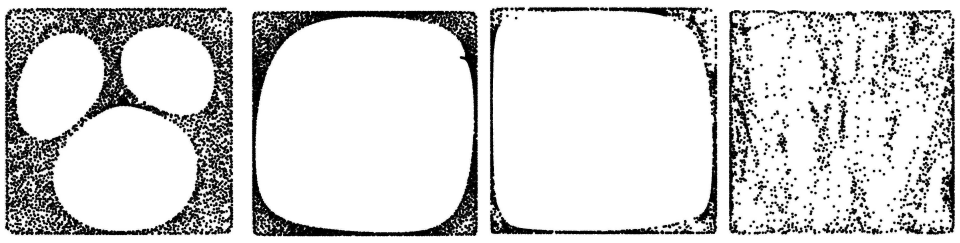


Fig. 10. From left to right, we have: (i) the true sensor locations and extracted combinatorial Delaunay complex; (ii) embedding of the combinatorial Delaunay complex; (iii) localization of all nodes by our algorithm; (iv) the results produced by MDS on all nodes in the network. The connectivity network is generated with unit disk graph model on nodes placed at perturbed grid points. First row: Cactus, 1692 nodes with average degree of 6.9. Second row: Ginger man, 2807 nodes with average degree of 10. Third row: Pretzel, 2993 nodes with average degree of 9.1. Fourth row: Smiley face, 2782 nodes with average degree of 9.5. Fifth row: Spiral in a box, 2910 nodes with average degree of 9.5. Sixth row: Square with a concave hole, 2161 nodes with average degree of 10.4.



(i) (ii) (iii) (iv)

Fig. 11. Rubberband algorithm results for (i) face (ii) spiral in a box (iii) square with a concave hole (iv) U shape.

of our algorithm is fairly stable for all kinds of shapes, but the performance of MDS depends a lot on the shape of the sensor field. We thus include here a few representative pictures in Figure 10.

Figure 10 (ii), (iii) shows the results of our algorithm for both the embedding of the combinatorial Delaunay complex and the localization result for all nodes. We put on the side the embedding results by MDS in Figure 10 (iv). MDS gives reasonable results for some cases (the 1st and 2nd example) but performs quite poorly when the real network has curved pieces (like spirals), and may even introduce an incorrect global flip, as in the 5nd and 6th examples. For a qualitative measure, We have computed the average distance error between the true location and our localization result and that of MDS, scaled by the communication range[7]. In all cases we are consistently better. In some cases when MDS does not produce the correct network layout, we are $4 \sim 7$ times better as shown in Table I.

Topology	concave	face	man	pretzel	spiral	cactus	star
Our Alg	1.88	0.91	1.94	0.95	1.11	2.39	2.16
MDS	4.42	2.78	3.24	1.45	7.10	2.82	3.24

Table I. Average location error, scaled by communication range.

4.3.2 *Influence of network communication models.* We tested our algorithm on different communication models. The observation is that the embedding result heavily depends on the performance of the boundary detection algorithm. If the boundary detection algorithm faithfully detected the network boundary, the embedding result is satisfactory as well. If the boundaries detected have local deficiencies, then the embedding may have local errors or flips. We show some representative cases in Figure 12. Figure 12 (i) and (ii) show what happens when a percentage of the links are broken. In (i) a fraction q of the edges in the unit disk graph, randomly selected, are deleted, for $q = 0.1$ and (ii) $q = 0.2$. In (iii) a quasi-UDG model is used: for two nodes whose distance d is between α and 1, there is an edge with probability $(1\text{-}d)/(1\text{-}\alpha)$. If $d < \alpha$, there must be an edge between them. α

[7]For alignment, we take three arbitrary landmarks and compute a rotation matrix for both results.

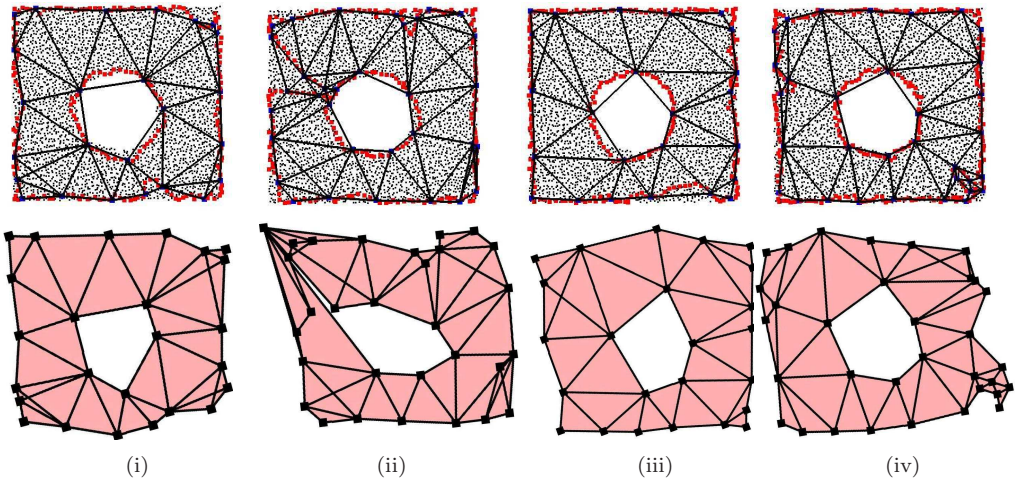


Fig. 12. Embedding the landmarks under challenging network conditions. The first row shows the ground truth; the second row our embedding of the landmark nodes. From left to right the models depicted are (i) 3443 nodes, avg. degree 10.66. only keep α edges and delete (1-α) edges randomly. α= 0.9. (ii) 3443 nodes, avg. degree 11.95. $\alpha = 0.8$ (iii) 3443 nodes, avg. degree 9.58. quasi-UDG model: We assume that for two nodes whose distance d is between α and 1, there is an edge with probability (1-d)/(1-α). If $d < \alpha$, there must be an edge between them. $\alpha = 0.8$ (iv) 3443 nodes, avg. degree 7.57. $\alpha = 0.6$.

= 0.8 in this case. In (iv), we use a quasi-UDG model with $\alpha = 0.6$. As you can see (ii) and (iv) give poor results. The problem in these cases is that the network boundary was not detected accurately. Whenever the boundary deviates from the real network boundary, we discovered that the embedding of the Delaunay triangles may incur local flips (such as the left top corner in (ii) and the right bottom corner in (iv)), as the information carried by the landmarks and the Delaunay triangles on these landmarks is now misleading.

4.3.3 *Influence of node density.* As node density goes higher, the performance of our algorithm improves. There are two reasons for this. One is that the boundary detection algorithm works better with higher node density. The second is that the hop-count distance between nodes is a better approximation of the geodesic distance between them.

The simulations in Figure 13 show the results of networks having increasingly denser nodes from left to right with the same communication range. Networks with higher density normally perform better than lower density networks. Specially, if the average degree is below 7, the boundary detection step fails to faithfully recover the boundary causing the rest of the algorithm performs not good as well.

4.3.4 *Influence of landmark density.* The theoretical results in the previous section gives a lower bound on the landmark density to ensure the rigidity of the Delaunay complex. One can certainly select much more landmarks than that. In general, a higher density of landmarks may allow for a slightly better embedding

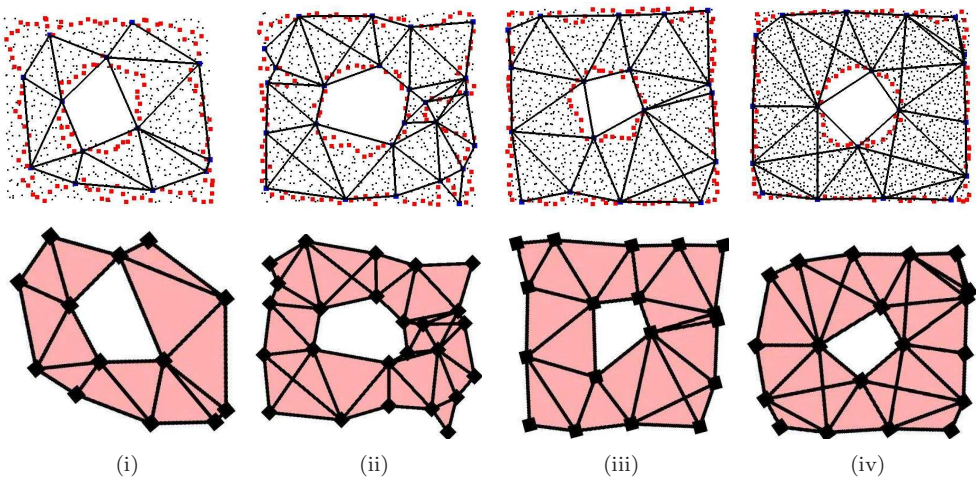


Fig. 13. Effect of node density/average degree on the embedding, the node densities increase from left to right and the communication ranges are the same for all networks. (i) 677 nodes, avg. degree 5.59 (ii) 840 nodes, avg. degree 6.56 (iii) 1162 nodes, avg. degree 9.2 (iv) 1740 nodes, avg. degree 14.57.

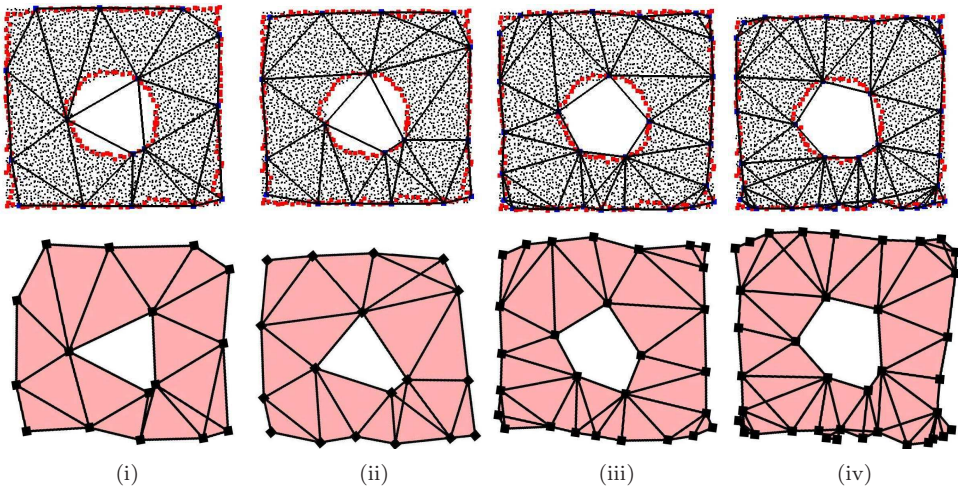
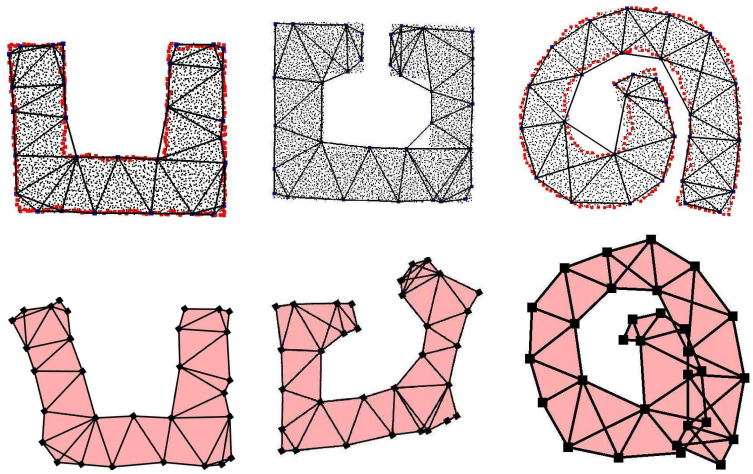


Fig. 14. Effect of landmark density. All figures with 3443 nodes and avg. degree 11.95. (i) decrease the number of landmarks (ii) standard number of landmarks as we described in algorithm section (iii) increase the number of landmarks (iv) increase the number of landmarks more

of the network since bends and corners of the network can be captured more accurately. With a very sparse set of landmarks the distance between 2 neighboring landmarks can be grossly exaggerated because the multi-hop path may need to get around a corner. But a denser set of landmarks means that the mass spring embedding of the Delaunay complex runs on a larger set, increasing the computation

and communication cost of the algorithm. As shown in Figure 14, the result of the algorithm is fairly stable with different landmark density. Thus the benefit of using a denser set of landmarks may not outweigh the increased cost of doing so.



(i) (ii) (iii)

Fig. 15. Possible error accumulation in networks with an elongated shape. In column (i) 3297 nodes, avg. degree 3297. We show a U-shaped graph properly embedded with minor distortion due to the use of hop-count distances. In (ii), 5028 nodes, avg. degree 14.9. The embedded network with a 'C' shape endures higher distortion. In (iii), 3910 nodes, avg. degree 15. Error accumulation causes the spiral to overlap on itself.

4.3.5 *Error accumulation.* Recall that the algorithm uses the hop count distance between landmarks to approximate their geodesic distance. Thus we may observe error accumulation in the embedding when the network has an elongated shape as shown in Figure 15. In these examples, the embedded shape is distorted and may have self-overlap (as in example (iii)), due to error accumulation.

4.4 Further discussion

MDS. Multidimensional scaling is a standard statistical approach that takes the all pairs proximity and recovers a 2D embedding of the vertices with linear projection methods such as principle component analysis (PCA). To better understand why MDS introduces incorrect flips, the intuition behind it is that the network hole causes the hop count distances to be not necessarily a good estimation of the Euclidean distance of the nodes. For example, the node at the tip of the spiral has a fairly long network distance to the opposite node 'across the lake'. MDS has no way of distinguishing this imprecise and misleading measurements from other good distance estimates. In fact, the misleading measurements seem to 'outweigh' the good measurements and MDS eventually chooses to flip the spiral over. Our other examples also show that the MDS tends to enlarge the hole in the middle. Another

limitation is that MDS behaves more or less like a blackbox and it is not easy to interpret the results and not to mention improving it.

On a different note, we remark that using multi-dimensional scaling on the shortest path distance matrix in a unit-disk graph setting is essentially the same algorithm as in Isomap [Tenenbaum et al. 2000], proposed by Tenenbaum, de Silva and Langford, for non-linear dimension reduction for high-dimensional data embedded in a low dimensional manifold. The famous result tested in Isomap is a 2D swiss roll shape manifold in 3D. With shortest path distance metric instead of the Euclidean metric in the ambient space, Isomap is able to 'flatten up' the swiss roll and recover the non-linear manifold. If the points are embedded on a 2D manifold but with possibly holes, i.e., a slice of Swiss cheese rolled up in 3D, our algorithm will recover a much more faithful representation of the unfolded 2D manifold. The fundamental idea here of using carefully selected short distances and patching up the local simplices suggests a generic rule of recovering the inherent topology and geometry of data points in an ambient space. This is one direction we will explore further. In a general setting, it requires both the understanding of topological features inherent in the geodesic distances and rigidity results in higher dimensions, both of which are not trivial.

Graph rigidity. The theory of graph rigidity in 2D has been relatively well understood. For example, there is a combinatorial condition, the Laman condition, to characterize graphs that are generically rigid. There is also an efficient algorithm, the pebble game [Jacobs and Hendrickson 1997], to test whether a graph is generically rigid in time $O(n^2)$. Similarly, both a combinatorial characterization of globally rigid graphs and polynomial algorithms for testing such graphs are known [Hendrickson 1992; Berg and Jordán 2003]. It is however not trivial to apply these rigidity results in the development of efficient localization algorithms. Given a graph with the edge lengths specified, finding a valid graph realization in $\mathbb{R}^d$ for a fixed dimension d is a NP-complete problem [Aspnes et al. 2004; Badoiu et al. 2004; Saxe 1979]. Even if we know a graph is globally rigid in 2D, there is no known efficient algorithm to find the realization of the graph in 2D with given edge lengths.

The pioneer work of using rigidity theory in network localization [Eren et al. 2004; Goldenberg et al. 2005; Goldenberg et al. 2006; Moore et al. 2004; Anderson et al. 2007; So and Ye 2005; Biswas and Ye 2004] focuses on identifying special graphs that do admit efficient localization algorithms. We briefly explain these ideas here and compare with our approach. The first idea is to use trilateration graphs [Eren et al. 2004; Goldenberg et al. 2005; Goldenberg et al. 2006; Moore et al. 2004]. A trilateration graph is defined recursively. It is either a triangle or a trilateration graph with a trilateration extension, defined as adding an additional vertex with three edges to existing vertices. If the network contains a trilateration graph, one can exhaustively search for the 'seed' triangle in the graph and greedily find the trilateration extensions. Thus an incremental algorithm can be adopted to find the realization of the network. A trilateration graph is a stronger condition than global rigidity (i.e., there are globally rigid graphs that are not trilateration graphs). The second idea is to examine *d-uniquely localizable graphs*. A graph with known edge lengths is called uniquely d-localizable if there is a unique realization

of the graph in $\mathbb{R}^d$ and there is no non-trivial realizations in $\mathbb{R}^k$ with $k > d$. For example, a generic simplex of $d + 1$ vertices is uniquely d-localizable. For uniquely d-localizable graphs, So and Ye [So and Ye 2005; Biswas and Ye 2004] has shown that a semi-definite program is able to find the realization. It is not known whether d-localizability is a generic property and it is not clear whether there is a combinatorial characterization of graphs that are d-localizable. Both approaches require that network has sufficiently many edges to be globally rigid.

Comparatively, we focus on the global rigidity of the combinatorial Delaunay complex, that has high-order topological structures than graphs. The combinatorial Delaunay *complex* is globally rigid but the combinatorial Delaunay *graph* is not necessarily globally rigid. Different from the graph rigidity approach, this algorithm does not require explicitly that the network to be embedded is globally rigid. This sheds some light on solving the network localization problem when the network is (uniformly) sparse but not rigid, such as a grid-like network with punched holes. Our current algorithm does not work well in the case of extremely low density networks because the boundary detection algorithm fails to find the network boundary effectively. In future work we plan to remove the dependency of boundary detection step in the algorithm and hope to apply it in localizing low-density non-rigid networks.

5. CONCLUSION

In this paper we proposed an anchor-free localization algorithm for large-scale sensor deployment with holes and complex shape. The novelty of our localization scheme is to extract high-order topological information to solve the notoriously difficult problem of resolving flip ambiguities. Geometric information of sensor nodes (e.g. node locations) has been recognized as an important character in sensor networks. The global topology of the sensor field is shown in this paper to be helpful in recovering the network geometry.

ACKNOWLEDGMENTS

A preliminary version of this paper has appeared in Proc. of the 27th Annual IEEE Conference on Computer Communications (INFOCOM'08). This work is supported by NSF CAREER Award CNS-0643687. We thank Alexander Kröller, Joe Mitchell, Rik Sarkar, Meera Sitharam, Xianjin Zhu for helpful discussions on related issues.

REFERENCES

Amenta, N., Bern, M., and Eppstein, D. 1998. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing 60*, 125–135.

Anderson, B. D. O., Belhumeur, P. N., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., and Yang, Y. R. 2007. Graphical properties of easily localizable sensor networks. *Wireless Networks*.

Aspnes, J., Goldenberg, D., and Yang, Y. R. 2004. On the computational complexity of sensor network localization. In *The First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*. 32–44.

Badoiu, M., Demaine, E. D., Hajiaghayi, M. T., and Indyk, P. 2004. Low-dimensional embedding with extra information. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*. 320–329.

Bateni, M., Demaine, E. D., Hajiaghayi, M., and Moharrami, M. 2007. Plane embeddings of planar graph metrics. *Discrete Comput. Geom. 38,* 3, 615–637.

Berg, A. R. and Jordán, T. 2003. A proof of connelly's conjecture on 3-connected generic cycles. *J. Comb. Theory B 88,* 1, 17–37.

Biswas, P. and Ye, Y. 2004. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the third international symposium on Information processing in sensor networks*. 46–54.

Bott, R. and Tu, L. 1982. *Differential Forms in Algebraic Topology*. Springer-Verlag.

Bruck, J., Gao, J., and Jiang, A. 2005. MAP: Medial axis based geometric routing in sensor networks. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. 88–102.

Cabello, S., Demaine, E. D., and Rote, G. 2007. Planar embeddings of graphs with specified edge lengths. *Journal of Graph Algorithms and Applications 11,* 1, 259–276.

Carlsson, G. and de Silva, V. 2004. Topological approximation by small simplicial complexes. In *Proceedings of the Symposium on Point-Based Graphics*.

de Silva, V. 2003. A weak definition of Delaunay triangulation. Tech. rep., Stanford University. October.

Edelsbrunner, H. 2001. *Geometry and Topology for Mesh Generation*. Cambridge Univ. Press.

Elson, J. 2003. Time synchronization in wireless sensor networks. Ph.D. thesis, University of California, Los Angeles.

Eren, T., Goldenberg, D., Whitley, W., Yang, Y., Morse, S., Anderson, B., and Belhumeur, P. 2004. Rigidity, computation, and randomization of network localization. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*. Vol. 4. 2673–2684.

Fang, Q., Gao, J., Guibas, L., de Silva, V., and Zhang, L. 2005. GLIDER: Gradient landmark-based distributed routing for sensor networks. In *Proc. of the 24th Conference of the IEEE Communication Society (INFOCOM)*. Vol. 1. 339–350.

Fang, Q., Gao, J., and Guibas, L. J. 2006. Landmark-based information storage and retrieval in sensor networks. In *The 25th Conference of the IEEE Communication Society (INFOCOM'06)*. Vol. 1. 339–350.

Fekete, S. P., Kaufmann, M., Kröller, A., and Lehmann, N. 2005. A new approach for boundary recognition in geometric sensor networks. In *Proceedings 17th Canadian Conference on Computational Geometry*. 82–85.

Fekete, S. P., Kröller, A., Pfisterer, D., Fischer, S., and Buschmann, C. 2004. Neighborhood-based topology recognition in sensor networks. In *ALGOSENSORS*. LNCS, vol. 3121. 123–136.

Fruchterman, T. M. J. and Reingold, E. M. 1991. Graph drawing by force-directed placement. *Softw. Pract. Exper. 21,* 11, 1129–1164.

Funke, S. 2005. Topological hole detection in wireless sensor networks and its applications. In *DIALM-POMC '05: Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing*. 44–53.

Funke, S. and Klein, C. 2006. Hole detection or: "how much geometry hides in connectivity?". In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*. 377–385.

Funke, S. and Milosavljević, N. 2007a. Guaranteed-delivery geographic routing under uncertain node locations. In *Proceedings of the 26th Conference of the IEEE Communications Society (INFOCOM'07)*. 1244–1252.

Funke, S. and Milosavljević, N. 2007b. Network sketching or: "how much geometry hides in connectivity? - part II". In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 958–967.

Ganeriwal, S., Kumar, R., and Srivastava, M. B. 2003. Timing-sync protocol for sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. 138–149.

Gao, J., Guibas, L., Oudot, S., and Wang, Y. 2008. Geodesic delaunay triangulation and witness complex in the plane. In *Proc. 18th ACM-SIAM Sympos. on Discrete Algorithms*. 571–580.

Goldenberg, D., Bihler, P., Cao, M., Fang, J., Anderson, B. D., Morse, A. S., and Yang, Y. R. 2006. Localization in sparse networks using sweeps. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. 110–121.

Goldenberg, D., Krishnamurthy, A., Maness, W., Yang, Y. R., Young, A., Morse, A. S., Savvides, A., and Anderson, B. 2005. Network localization in partially localizable networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*. Vol. 1. 313–326.

Graver, J. E., Servatius, B., and Servatius, H. 1993. *Combinatorial Rigidity*. Graduate Studies in Math., AMS.

Hendrickson, B. 1992. Conditions for unique graph realizations. *SIAM J. Comput. 21,* 1, 65–84.

Howard, A., Mataric, M., and Sukhatme, G. Relaxation on a mesh: a formalism for generalized localization. In *In Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 1055–1060.

Jacobs, D. J. and Hendrickson, B. 1997. An algorithm for two-dimensional rigidity percolation: the pebble game. *J. Comput. Phys. 137,* 2, 346–365.

Kamada, T. and Kawai, S. 1989. An algorithm for drawing general undirected graphs. *Inf. Process. Lett. 31,* 1, 7–15.

Kobourov, S. G., Efrat, A., Forrester, D., and Iyer, A. 2006. Force-directed approaches to sensor network localization. In *8th Workshop on Algorithm Engineering and Experiments (ALENEX)*.

Kröller, A., Fekete, S. P., Pfisterer, D., and Fischer, S. 2006. Deterministic boundary recognition and topology extraction for large sensor networks. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*. 1000–1009.

Laman, G. 1970. On graphs and rigidity of plane skeletal structures. *J. Engineering Math.* 4, 331–340.

Moore, D., Leonard, J., Rus, D., and Teller, S. 2004. Robust distributed network localization with noisy range measurements. In *SenSys'04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. 50–61.

Priyantha, N. B., Balakrishnan, H., Demaine, E., and Teller, S. 2003. Anchor-free distributed localization in sensor networks. Tech. Rep. TR-892, MIT LCS. April.

Rao, A., Papadimitriou, C., Shenker, S., and Stoica, I. 2003. Geographic routing without location information. In *Proceedings of the 9th annual international conference on Mobile computing and networking*. 96–108.

Savvides, A., Han, C.-C., and Strivastava, M. B. 2001. Dynamic fine-grained localization in *ad-hoc* networks of sensors. In *MobiCom '01: Proceedings of the 7th ACM Annual International Conference on Mobile Computing and Networking*. 166–179.

Saxe, J. B. 1979. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings of the 17th Allerton Conference in Communications, Control and Computing*. 480–489.

Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. P. J. 2003. Localization from mere connectivity. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. 201–212.

So, A. M.-C. and Ye, Y. 2005. Theory of semidefinite programming for sensor network localization. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. 405–414.

Tenenbaum, J., de Silva, V., and Langford, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290,* 22 (December), 2319–323.

Tutte, W. T. 1963. How to draw a graph. *Proceedings London Mathematical Society 13,* 52, 743–768.

Wang, Y., Gao, J., and Mitchell, J. S. B. 2006. Boundary recognition in sensor networks by topological methods. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. 122–133.

Zhu, X., Sarkar, R., and Gao, J. 2007. Shape segmentation and applications in sensor networks. In *Proceedings of the 26th Conference of the IEEE Communications Society (INFOCOM'07)*. 1838–1846.

6. APPENDIX

6.1 Proofs in section 2.1

Observation 2.2. *The inner medial axis of $\mathcal{R}$ measured in terms of Euclidean distance is the same as that measured in terms of geodesic distance.*

Proof. Take the maximum size ball centered at a point p on the medial axis under Euclidean distance measure. This ball touches two or more points on the boundary and has no boundary points in its interior. Thus the geodesic distances from p to the tangent points are the same as the Euclidean distances. In other words, a point p is on the medial axis under the Euclidean distance is also on the medial axis under the geodesic measure.

On the other hand, take a maximum size ball centered at a point p on the medial axis under the geodesic distance measure and its tangent points on $\partial\mathcal{R}$. We argue that the geodesic shortest path from p to its tangent point must be a straight line. If otherwise it can only bend at a point q on the boundary $\partial\mathcal{R}$. This means q is a closer boundary point than the tangent point, which contradicts with the assumption. Thus the point p is also on the medial axis under the geodesic distance measure. □

Next we prove an important Lemma about the inner local feature size. This Lemma and its proof are motivated by [Amenta et al. 1998] and will be useful for the proofs in Subsection 2.3.

Lemma 6.1. *Given a disk B containing at least two points on $\partial\mathcal{R}$, for each connected component of $B \cap \mathcal{R}$, either it contains a point on the inner medial axis, or its intersection with $\partial\mathcal{R}$ is connected.*

Proof. We take one connected component C of $B \cap \mathcal{R}$ and assume that it does not contain a point on the inner medial axis and intersects $\partial\mathcal{R}$ in two or more connected pieces. Now we take a point u in C but u is not on $\partial\mathcal{R}$. Now take u's closest point on $C \cap \partial\mathcal{R}$. If the closest point is not unique, then u is on the inner medial axis and we have a contradiction. Now the closet point p stays on one connected piece of $C \cap \partial\mathcal{R}$. We take u's closest point on a different piece of $C \cap \partial\mathcal{R}$, denoted as q. See Figure 16. Now as we move a point x from u to q along the geodesic path between u and q, x's closest point on $C \cap \partial\mathcal{R}$ starts with p and eventually becomes q. So at some point x the closest point changes. That point x is on the inner medial axis. This leads to a contradiction, and hence the claim is true. □

6.2 Proofs in section 2.3

Observation 2.7. *Two Voronoi vertices connected by a Voronoi edge correspond to two Delaunay triangles sharing an edge.*

Proof. Recall that each Voronoi vertex x certifies a Delaunay triangle of three landmarks u, v, w. First we argue that the points on the Voronoi edge connecting

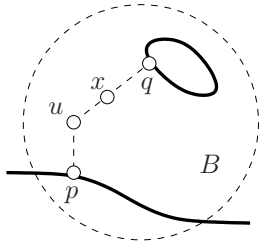


Fig. 16. Each connected component of $B \cap \mathcal{R}$ either contains a point on the inner medial axis or its intersection with $\partial\mathcal{R}$ is connected.

Voronoi vertices x and y must have their two closest landmarks among u, v, w. Certainly if one point on the Voronoi edge has one of its closest landmark to be p and p is not any of u, v, w, then this point is a Voronoi vertex. Without loss of generality, we assume that y has three closest landmarks u, v, z. Thus the corresponding Delaunay triangles of x, y are $\triangle uvw$ and $\triangle uvz$ sharing an edge uv. □

Observation 2.8. *For any two adjacent landmarks u, v on the same boundary cycle, there must be a Voronoi vertex inside $\mathcal{R}$ whose closest landmarks include u, v.*

Proof. We take two adjacent landmarks u, v and consider the set of points in $\mathcal{R}$ with equal distance from u, v. The mid-point on the geodesic path connecting u, v, denoted by x, is at an equal distance from u, v. We take a disk through u, v centered at x and move the disk while keeping it through u, v. Its center will trace a curve called $C(u, v)$ with all the points on $C(u, v)$ having equal distances from u, v. $C(u, v)$ has two endpoints p, q with q on the boundary segment in between u, v and p also on the boundary. Take $r = d(p, u) = d(p, v)$. See Figure 17.

We claim that there must be a Voronoi vertex on $C(u, v)$ that involves u, v and we prove this claim by contradiction. Otherwise, p's two closest landmarks are u, v — the ball $B_r(p)$ centered at p with radius r contains no other landmark inside. We take $r^- = r - \varepsilon$ with $\varepsilon \to 0$. Thus $B_{r^-}(p)$ contains no landmark. Now we see that this will violate the sampling condition if we can show that there is a point on the inner medial axis inside $B_{r^-}(p)$ (meaning that $r^- \geq ILFS(p)$).

We take the connected component of $B_{r^-}(p) \cap \mathcal{R}$ that contains the curve $C(u, v)$, denoted by F. By Lemma 6.1, if F does not contain a point on the inner medial axis, then its intersection with the boundary $\partial\mathcal{R}$ is connected. Now we do a case analysis depending on how the boundary curve goes through u and v. In Figure 17 (i) & (ii), the ε-neighborhood of the boundary at u, v also intersects $B_{r^-}(p) \cap \mathcal{R}$. In (i), $F \cap \partial\mathcal{R}$ has two connected pieces, thus leading to a contradiction. In (ii), the boundary between u, v through p is completely inside $B_{r^-}(p)$, which has no other landmark inside. In this case there are only 2 landmarks, namely u, v, on the boundary cycle containing p. This contradicts our sampling condition.

If the boundary at v (or u, or both) is only tangent to $B_{r^-}(p) \cap \mathcal{R}$ (meaning that $B_{r^-}(p)$ does not contain any ε-neighborhood of v, see Figure 17 (iii) & (iv)), we argue that F contains a point on the inner medial axis. To see that, we take the ball $B_r(p)$ tangent at v with v's ε-neighborhood outside the ball. Now we shrink it

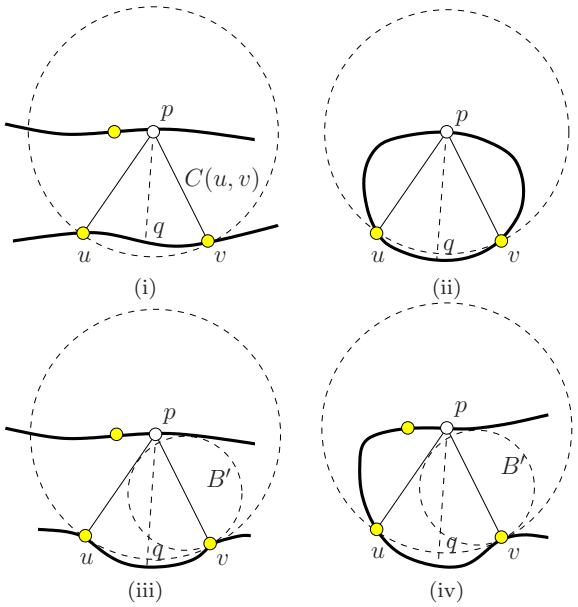


Fig. 17. u, v are two adjacent landmarks. The point p on the boundary has its closest landmarks as u, v. (i)-(iv) four possible cases.

while keeping it tangent to v until it is tangent to two points on the boundary of F. Now the center of the small ball B' is on the inner medial axis, which is inside $B_{r-}(p)$. Thus we have the contradiction. The claim is true. □

Lemma 2.9. *If there is a continuous curve C that connects two points on the boundary $\partial\mathcal{R}$ such that C does not contain any point on Voronoi edges, then C cuts off a topological 1-disk of $\partial\mathcal{R}$ with at most one landmark inside.*

Proof. Without loss of generality we assume that C has no other boundary points in its interior. Assume C connects two points p, q on the boundary. Since C does not cut any Voronoi edges, C must stay completely inside the Voronoi cell of one landmark say u. Without loss of generality assume that u is to the right of boundary point q. See Figure 18(i). Now the boundary of Voronoi cell of u is

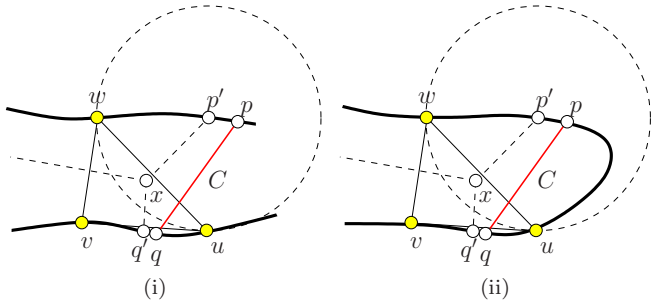


Fig. 18. (i) C is inside the Voronoi cell of landmark u to the right of C. (ii) the curve C cuts off a segment of $\partial\mathcal{R}$ with no other landmark inside.

partitioned by the curve C, with one part completely to the left of C. Consider one of the intersections between the Voronoi cell boundary of u with the region boundary $\partial\mathcal{R}$, say p'. We consider the ball $B_r(p')$ with $r = d(p', u)$. The point p' has two closest landmark, with one of them as u and the other to the left of C, denoted as w. Now, this ball cannot contain any other landmark besides u, w. We argue by Lemma 6.1 that the component of $B_r(p') \cap \mathcal{R}$ containing p' intersects $\partial\mathcal{R}$ in a connected piece. Otherwise $B_r(p')$ contains a point on the inner medial axis, which means $r > ILFS(p')$. Thus by the sampling condition there must be a landmark inside $B_r(p')$.

Now, since the component of $B_r(p') \cap \mathcal{R}$ containing p' intersects $\partial\mathcal{R}$ in a connected piece, this intersection is a continuous segment between u and w on $\partial\mathcal{R}$, completely inside $B_r(p')$, by using the same argument as in the previous lemma; see Figure 18 (ii). In this case, the curve C cuts off a segment of $\partial\mathcal{R}$ with at most one landmark inside. The claim is true. □

View publication stats