Discovery of Sensor Network Layout using Connectivity Information

Sol Lederer* Yue Wang* Jie Gao*

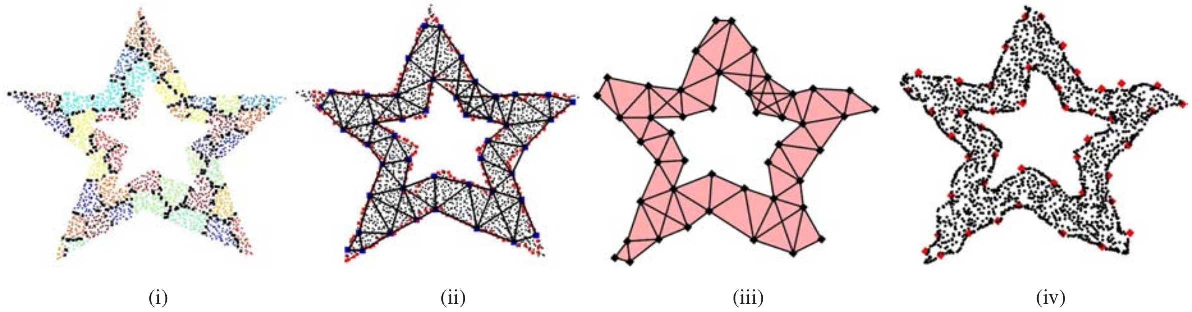


Figure 1. Discovery of the network layout (a double star shape). The number of nodes in this example is 2171 and the average node degree is 10. (i) The Voronoi cells of the landmarks (black nodes are on the Voronoi edges); (ii) The Delaunay edges extracted from the Voronoi cells of the landmarks; (iii) Our embedding result of the extracted Delaunay complex; (iv) Our localization result of the entire network.

Abstract

We propose a distributed algorithm to discover and recover the layout of a large sensor network having a complex shape. As sensor network deployments grow large in size and become non-uniform, localization algorithms suffer from "flip" ambiguities—where a part of the network folds on top of another while keeping all edge length measurements preserved. We explore the high-order topological information in a sensor field to prevent incorrect flips and accurately recover the shape of the sensor network. We select landmarks on network boundaries with sufficient density, construct the landmark Voronoi diagram and its dual combinatorial Delaunay complex on these landmarks. The key insight is that when the landmarks are dense enough to capture the local geometric complexity, the combinatorial Delaunay complex is globally rigid and has a unique realization in the plane. An embedding by simply gluing the Delaunay triangles properly derives a faithful network layout, which consequently leads to a practical and sufficiently accurate localization algorithm. We prove the global rigidity of the combinatorial Delaunay complex in the case of a continuous geometric region. Simulations on discrete networks show surprisingly good results, while multi-dimensional scaling and rubberband representation perform poorly or not at all in recovering the network layout.

1 Introduction

The physical location of sensor nodes in a network is critical for network operation and data interpretation. Localization algorithms that find the locations of nodes through local communication and message passing have become an indispensable architecture component. Many localization algorithms have been proposed in the past few years [35, 37, 38, 31, 32, 39, 4, 23, 11, 22, 21, 30], yet there is still no universally recognized localization algorithm that produces accurate location information with small overhead [33, 41]. As sensor networks grow in popularity and size, retrieving the locations of the nodes becomes even more important and challenging. The difficulty in achieving accurate location information is not only due to the network scale and increase to both the communication and computation load, but also due to the fact that large deployments of sensor nodes are more likely to have uneven node distribution as obstacles and terrain variations inevitably come in to the picture. Although most localization algorithms work reasonably well for uniform and dense sensor deployments, they often run into serious trouble when the network layout has complicated geometric features. The prominent difficulty is the rigidity issue and the problem of resolving incorrect flips. To give an intuitive example, Figure 2 illustrates that with only network connectivity information (and/or distance information), one is unable to tell the "flip" of triangle $\triangle bcd$ relative to triangle $\triangle abc$ locally. When the network is large, this flipping ambiguity issue can be so severe that many optimization-based approaches easily get stuck at local minima corresponding to configurations far from the ground truth. A number of localization algorithms deal with the problem of rigidity by exploring the graph structure [11, 22, 21, 30]. These algorithms either require that the network is dense enough to guarantee global rigidity, or even more, guarantee the network is a tri-lateration graph (such that iterative trilateration method resolves the ambiguity of flips) [11, 22, 30]; or, when the network is sparse, record all possible configurations and prune incompatible ones whenever possible,

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

Dagstuhl Seminar Proceedings 07151
Geometry in Sensor Networks
http://drops.dagstuhl.de/opus/volltexte/2007/1114

which, in the worst case, can result in exponential space requirement [21].

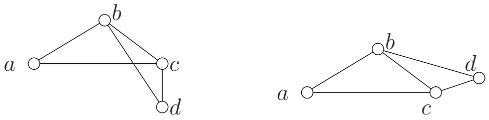


Figure 2. A connectivity graph with two distinct embeddings having the same set of edge lengths.

Aside from localization algorithms, recently there is a growing interest in the study of global topology and geometry of a sensor field deployment. The focus is to identify large topological features (such as coverage holes) by detecting the network boundaries [17, 18, 15, 14, 28, 40]. The identification of these features immediately indicates points of insufficient network coverage [20]. In addition, they can also help with basic network operations such as routing and information discovery [12, 8, 19, 13]. There is still an unclosed gap in the loop, as Funke and Milosavljevic spotted [19], in that the identification of boundary nodes still has not produced a global picture of the sensor field layout. Or, we know how many holes there are and which are the boundary nodes but we have no idea how they are laid out in the domain.

In this paper we are interested in the discovery and recovery of the network layout, which is of great help for large-scale network localization and also interesting on its own. The question we ask is: with connectivity information only, can we recover the global layout of the network? This is quite a challenging task due to the global nature of the problem. Existing approaches that use centralized global optimization techniques such as multi-dimensional scaling (MDS) or semi-definite programming [5, 39] are both computationally intensive and unsatisfactory quality-wise (see Figure 4 left). In addition, recovering the correct global layout is the most critical component in localization and also of great importance for network organization and operation. For example, greedy routing with imperfect geographical locations that differ from their true locations can still achieve high success delivery rate [36], if there are no global flips. To give an intuitive feeling, Figure 3 shows the real deployment of sensor nodes (left) and two possible embedding results that may have similar quality measured by the absolute location error. The middle one that does capture the most essential feature of the network layout (5 point star with no hole), is far better than the more devastating embedding to the right, in which part of the network incorrectly folds on top of the other.

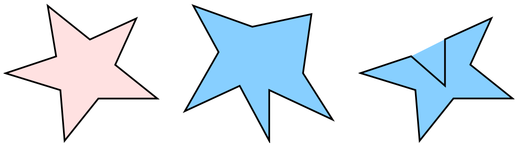


Figure 3. Left to right: the ground truth; one possible embedding; a more devastating embedding.

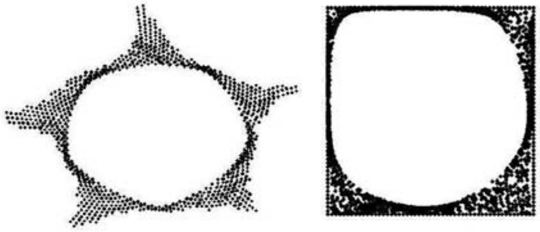


Figure 4. Embedding of the double star. Left: multi-dimensional scaling; Right: rubberband representation.

One step towards the eternal goal of recovering the network layout is to use the rubberband embedding, by Rao *et al.* in [36] and by Funke and Milosavljevic in [19]. 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. 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 4 (right). In the literature [36, 19] the rubberband representation is mainly used in extracting the sensor network topology and assigning virtual coordinates to the nodes for geographic routing purposes.

In this paper we propose an efficient algorithm that recovers the network layout with surprising accuracy. We explore the fact that the sensor nodes are embedded in the plane (or with terrain variation, on a 2-manifold surface). When there are incorrect global folds, some nodes with a large minimum hop distance in the network are brought closer than what it should be. Our layout discovery algorithm is quite simple and uses only the network connectivity information. We take samples on the network boundaries (both the outer boundary and inner hole boundaries) with sufficient density and denote select nodes as *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. The Delaunay graph, 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 such that they do not share interior points (the case in Figure 2 left cannot happen). 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. In this paper we make the above intuition rigorous. We prove in the case of a continuous geometric domain that when the landmarks are sufficiently dense (with respect to the local feature size measuring the local geometric complexity), the induced Delaunay graph is rigid. In addition, the Delaunay complex (in terms of simplicial complex, to be defined later) 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 the way to embed them relative to each other removes a major hurdle towards the recovery of the network layout. In this paper we use a simple 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 counts are inevitably noisy, we use mass-spring relaxation to improve the quality of the embedding and balance the error distribution. The final result on the double star case is shown in Figure 1 (iii). It is apparent that our result is much better than those produced by MDS or rubberband representation and very close to the ground truth.

As the recovery of the network layout when the nodes have non-uniform distribution is the major challenge in anchor-free localization algorithms (i.e., no nodes have any location information, the goal is to discover the relative node positioning up to a global rigid motion), the results from the layout algorithm can be used to localize the rest of the nodes easily. With the landmarks localized, each additional node can localize itself by using trilateration with its hop count distances to 3 or more landmarks. Again we apply mass-spring relaxation to further improve the localization result. The final localization result of the double star example is shown in Figure 1 (iv).

The outline of the paper is as follows. In Section 2 we prove the rigidity of the Delaunay complex when landmarks are sufficiently dense in the case of a continuous domain. The reason we explain the theory first is to introduce notations and subtle issues due to degeneracies that will show up in the discrete case as well. Readers can also choose to read Section 3 first, in which we explain the algorithm for the discrete network. The discovery of the sensor layout, i.e., landmark selection and discovery of the Delaunay edges is done in a distributed way. The recovery algorithm, i.e., embedding of the Delaunay complex, is of an incremental nature and thus can also naturally implemented in a distributed way by message passing. We tested our algorithms under various deployment scenarios and compared the embedding result with the results by using multi-dimensional scaling and the rubberband embedding in Section 4.

2 Theoretical Foundations

2.1 Medial axis, local feature size and r-sample

We consider a geometric region $\mathcal{R}$ with boundary $\partial\mathcal{R}$ that consists of k cycles, one outer boundary and $k-1$ 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 $\mathcal{R}$. 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 (measured by Euclidean distance) on the boundary $\partial\mathcal{R}$.*

The medial axis of ∂R consists of two components, one part inside $\mathcal{R}$ and denoted as the *inner medial axis*, and the other part outside $\mathcal{R}$, denoted as the *outer medial axis*. See Figure 5 (i). In this paper we are more interested in 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. In a sensor network without location information, 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. 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.

Lemma 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. □

Now we are ready to explain how to measure the local geometric complexity of $\mathcal{R}$, which will consequently decide the sampling density. An example is shown in Figure 5.

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

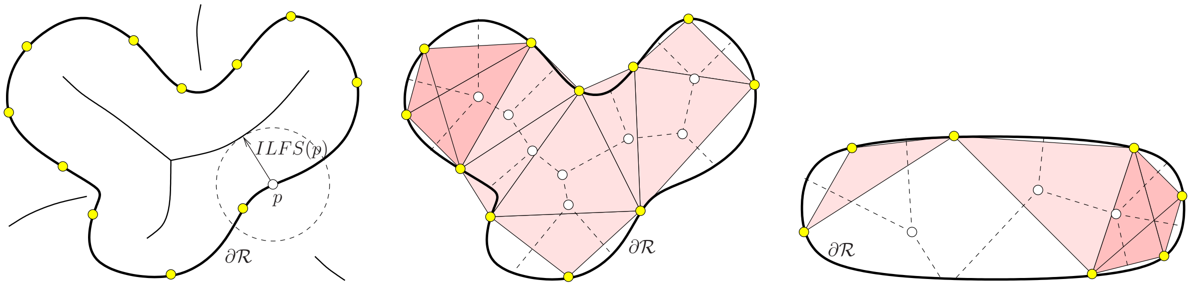


Figure 5. The region $\mathcal{R}$ with boundary shown in dark curves. (i) The medial axis and landmarks selected on the boundaries. Point $p \in \partial\mathcal{R}$ has a landmark within distance $ILFS(p)$. (ii) The Voronoi graph (shown in dashed lines) and the Delaunay graph/complex. (iii) When the set of landmarks is not an r-sample (with $r < 1$), the combinatorial Delaunay graph may be non-rigid.

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).*

Next we prove an important Lemma about the inner local feature size. This Lemma and its proof are motivated by [2].

Lemma 2.4. *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 6. 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 arrives at a contradiction and the claim is true. □

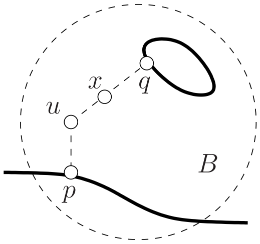


Figure 6. Each connected component of $B \cap \mathcal{R}$ either contains a point on the inner medial axis or its intersection with $\partial\mathcal{R}$ is connected.

Definition 2.5. *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 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 imply 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 [12]. 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. Rigorously,

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

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. An example is shown in Figure 5. We state some immediate observation about the Voronoi diagram and the corresponding combinatorial Delaunay graph below.

Observation 2.6. *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 (ii). We will capture these high-order features by defining the Delaunay complex in the notion of abstract simplicial complex [10]. Formally, a finite system A of finite sets is an *abstract simplicial complex* if $\alpha \in A$ and $\beta \subseteq \alpha$ implies $\beta \in A$. 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. The way to construct an abstract Delaunay complex is to take the *Cěch complex* of the Voronoi cells, defined below.

Definition 2.7. *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, thus the name of 'abstract' complex. We can talk about an embedding or realization of an abstract simplicial complex in a geometric space as a *simplicial complex* (with geometry). We give the definitions below. The whole point of this paper is to find the geometric realization of the abstract Delaunay complex extracted from a sensor field, to recover its global layout.

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 7 shows 0, 1, 2, 3-simplex in $\mathbb{R}^3$. The convex 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 $\operatorname{conv} \varphi(\alpha) \in K$. Of course the embedded ambient space 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 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 etc.

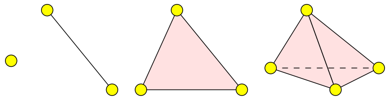


Figure 7. 0, 1, 2, 3-simplex in $\mathbb{R}^3$.

2.3 Global rigidity of combinatorial Delaunay complex

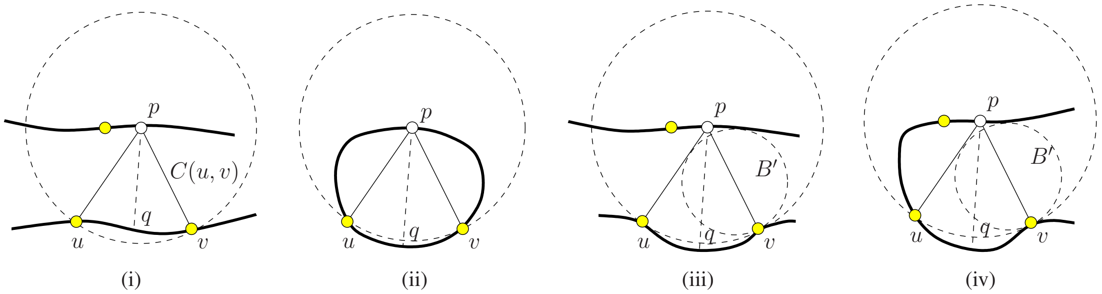
The property of the combinatorial Delaunay graph clearly depends on the selection of landmarks. The goal of this section is to show that with sufficiently dense landmarks — when there are at least 3 landmarks on each boundary cycle and they form an r-sample of $\partial\mathcal{R}$ with $r < 1$—the Delaunay graph is rigid (no continuous deformation possible if the edges are of fixed lengthes) and the Delaunay complex is globally rigid (it admits a unique realization). An example when the combinatorial Delaunay graph is not rigid due to insufficient sampling is shown in Figure 5 (iii). 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.

Lemma 2.8. *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 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. □

Lemma 2.9. *For any two adjacent landmarks u, v on the same boundary cycle, there must be a Voronoi vertex inside $\mathcal{R}$ that involves these two landmarks.*

Proof: We take two adjacent landmarks u, v and consider the set of points in $\mathcal{R}$ with equal distance from u, v. The



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

Figure 8. u, v are two adjacent landmarks. The point p on the boundary has its closest landmarks as u, v. (i)-(iv) four possible cases.

mid-point on the geodesic path connecting u, v, denoted by x, is 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 8.

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 $r^- \geq ILFS(p)$).

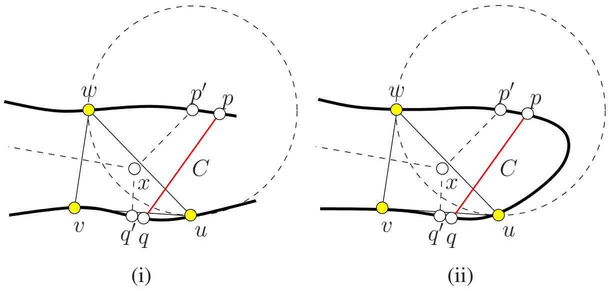
We take the connected component of $B_{r^-}(p) \cap \mathcal{R}$ that contains the curve $C(u, v)$, denoted by F. By Lemma 2.4, if F does not contain a point on the inner medial axis, then its intersection with the boundary $\partial\mathcal{R}$ is connected. Now we do a case analysis depending on how the boundary curve goes through u and v. In Figure 8 (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 8 (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 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 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 2.10. *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 no other 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 9(i). Now the boundary of Voronoi cell of u is 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 $r = d(p', u)$. 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 2.4 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 $B_r(p')$.



(i) (ii)

Figure 9. (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.

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 9 (ii). In this case, the curve C cuts off a segment of $\partial\mathcal{R}$ with no other landmark inside. The claim is true. □

Corollary 2.11. *The Voronoi graph $V(S)$ is connected.*

Proof: This follows immediately from Lemma 2.10, although Lemma 2.10 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. □

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 lengthes 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.

Theorem 2.12 (Laman condition [29]). *A graph G with n vertices is generically rigid*[1] *in 2 dimensions if and only if it contains a Laman graph G', which has $2n-3$ edges and every subset of k vertices spans at most $2k-3$ edges.*

Theorem 2.13. *The combinatorial Delaunay graph $D(S)$ is rigid.*

Proof: In this proof we assume without loss of generality that there is no degenerate case, i.e., four or more landmarks are not co-circular. Indeed we can perturb the locations of the landmarks slightly to eliminate the degeneracy. This will only remove edges from the the combinatorial Delaunay graph. Certainly with more Delaunay edges introduced by degeneracy the graph is still rigid.

From the Voronoi graph $V(S)$, we extract a subgraph V' that contains all the Voronoi vertices and the Voronoi edges that connect between these Voronoi vertices. Some Voronoi edges end at points on the boundary $\partial\mathcal{R}$ and we ignore those. By Corollary 2.11 this graph V' is connected. Now we find a spanning tree T in V' that connects all the 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 though 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 such that both of the corresponding landmarks 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 as above 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—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 embeddings. 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, find a mapping φ of the vertices in the plane such that any 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 [24, 3] show that a graph is globally rigid (uniquely realizable) in 2D under

[1] Intuitively, generic rigidity means that almost all (except some degenerate cases) realizations of the graph in the plane are rigid – one cannot continuous move the vertices, up to a global translation and rotation, without changing the lengths of the edges. In other words, the graph cannot be deformed continuously in the plane.

edge lengthes constraints if and only if it's tri-connected and is redundantly rigid. Thus all high-order simplices have unique embeddings 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, such that 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 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 10(i). Otherwise, w and p are embedded on the same side of uv.

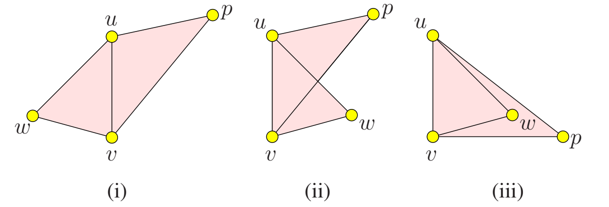


Figure 10. Two Delaunay triangles $\triangle uvw$ and $\triangle uvp$ sharing an edge. (i) is the only valid valid embedding with the two triangles not sharing any interior points.

Then either w is inside $\triangle uvp$ (as in Figure 10 (iii), or p is inside $\triangle uvw$, or two edges intersect at a non-vertex point (as in Figure 10 (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 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 2.14. *Under our landmark selection criterion, the combinatorial Delaunay complex $DC(S)$ has at most one embedding in the plane up to a global translation and rotation.*

2.4 A nasty case with special handling

Our sampling condition is unfortunately not sufficient to guarantee that the combinatorial Delaunay complex is homotopy equivalent[2] 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 $\mathcal{R}$. A bad example is shown in Figure 11. To see why this is bad, the Voronoi edge of 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. Thus there is no valid geometric realization in the plane without violating the properties of a simplicial complex.

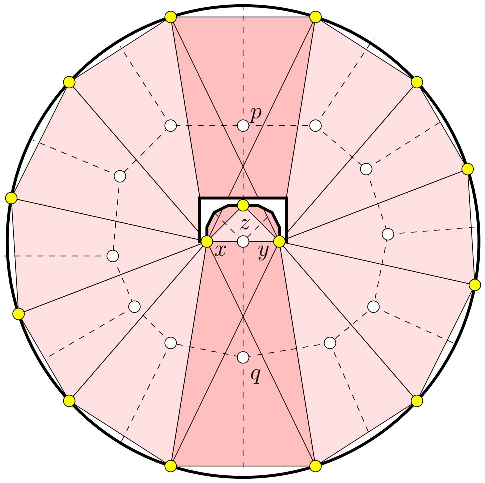


Figure 11. A nasty example with no valid embedding of the Delaunay complex.

We deal with this problem in two ways. As will be shown in the next section, we are able to detect that whether the Voronoi edge is connected or not. If it happens, we insert more landmarks. In particular, we include one of the Voronoi vertex with x, y as a landmark and continue doing this until the Voronoi edge between x, y is simply connected. Similarly, if three or more landmarks have two or more Voronoi vertices, then we kill all but one of them by including those as landmarks. As the following theorem shows, as long as the Voronoi edge/vertex set of any k landmarks is either empty or contractible[3], the homotopy equivalence is established.

Lemma 2.15. *If we have at least one landmark on each hole boundary, the Voronoi cell of a landmark u is simply connected (no holes).*

Proof: First we show that the Voronoi cell of a landmark u is connected. Any point q on the shortest path connecting u with one point $p \in V(u)$ also stay in $V(u)$. Otherwise say q has landmark v as its closest landmark, then v is a closer landmark to p than u.

[2] 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.

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

If the Voronoi cell is not simply connected, i.e., there is a hole entirely inside $V(u)$. This cannot happen as any landmark on the hole boundary, thus inside $V(u)$, will destroy the property of this Voronoi cell. □

Theorem 2.16. *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 $\mathcal{R}$.*

Proof: As the combinatorial Delaunay complex is the Cěch complex of the Voronoi cells, the theorem follows from the Cěch Theorem [7]. 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 $\mathcal{R}$. □

Another way to handle this is to embed the Delaunay complex anyway — the embedding theoretically violates the simplicial complex definition but in practice is perfectly fine. One thing we notice is that we do know how to embed the triangle $\triangle xyz$ 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 to 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 explain in this section the distributed algorithm to extract the combinatorial Delaunay complex as well as the embedding algorithm that recovers the network layout. Our algorithm only uses the connectivity information and does not require that the network model follow the unit disk graph assumption. We assume a large number of sensor nodes scattered in a geometric region, with nearby nodes communicating with each other directly. The algorithm basically realizes the landmark selection and embedding suggested in the previous section. Thus we will not re-iterate many things said already and instead focus on distributed implementation and robustness issues, for the geodesic distance is only approximated by the minimum hop count between two sensor nodes. We first outline the algorithm and explain each step in detail.

Detect network boundary. Nodes on the network boundaries are identified and connected into boundary cycles surrounding inner holes and the outer face by a boundary detection algorithm [40]. The inner medial axis is also identified during this process.

Select landmarks. 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. k landmarks are included in a Delaunay simplex if their Voronoi cells share a common node. See Figure 1(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.

(Optional) 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. Thus we have an anchor-free localization algorithm that is particularly attractive for sensor network of complex shape.

3.1 Detect network boundary

We use a distributed boundary detection algorithm that identifies nodes on both outer and inner boundaries and connects them into boundary cycles [40]. The boundaries of the sensor field can be used to generate the medial axis of the sensor field, defined as the set of nodes with at least two closest boundary nodes. To do that, the boundary nodes flood inward simultaneously. The flooding messages are suppressed by the hop count from the boundary nodes to improve the 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 network boundary is updated and the message is further forwarded. In this way each node learns its closest boundary node. The nodes at which the flooding frontiers collide are nodes on the inner medial axis.

3.2 Select landmarks

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 simultaneously with proper message suppression in a way similar with that in the previous subsection. Each boundary node learns its local feature size as the hop count to its closest node on the medial axis. Landmark selection is performed by a message traversing along the boundary cycles. 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.

In cases when the boundary detection algorithm [17, 18, 15, 14, 28] does not produce boundary cycles, 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.

3.3 Compute Voronoi diagram and combinatorial Delaunay complex

The landmark Voronoi diagram is computed in a distributed way as in [12]. 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. So 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. Due to the discreteness of hop count values, we re-define Voronoi vertices.

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. In this case, the closest landmark is called the* home landmark *of p. A* border node *is a node that is not an interior node.*

Figure 1 (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, or in other words, 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 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 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. Or, if the minimum hop count distances between neighboring landmarks are desired, one can let the messages initiated by the landmarks earlier 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.

3.4 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 2.14), 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 conflicting 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 [27, 25, 26, 16, 34] 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 as Figure 12 shows, 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 [27].

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.

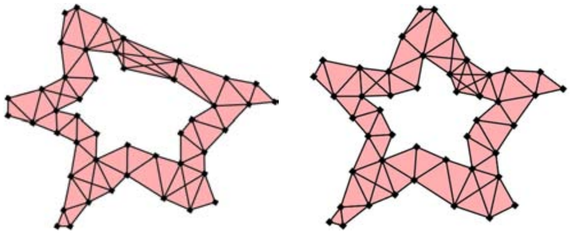


Figure 12. left: before the mass-spring relaxation algorithm is applied; Right: after mass-spring relaxation.

In a distributed environment the embedding of the Delaunay simplices can be done incrementally with message passing. 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 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.

3.5 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 [37]) by using the hop count estimation to 3 or more landmarks. 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 including multi-dimensional scaling and rubberband representation.

Multi-dimensional scaling (MDS). Multidimensional scaling has been used by Shang et al. [39] for sensor network localization with connectivity information only. 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). However since only rough hop-count distances are known, MDS is unsatisfactory in practice. As our simulations will show, MDS has trouble capturing a twist within the graph, making a long narrow graph not differentiable from a spiral-shaped graph. In

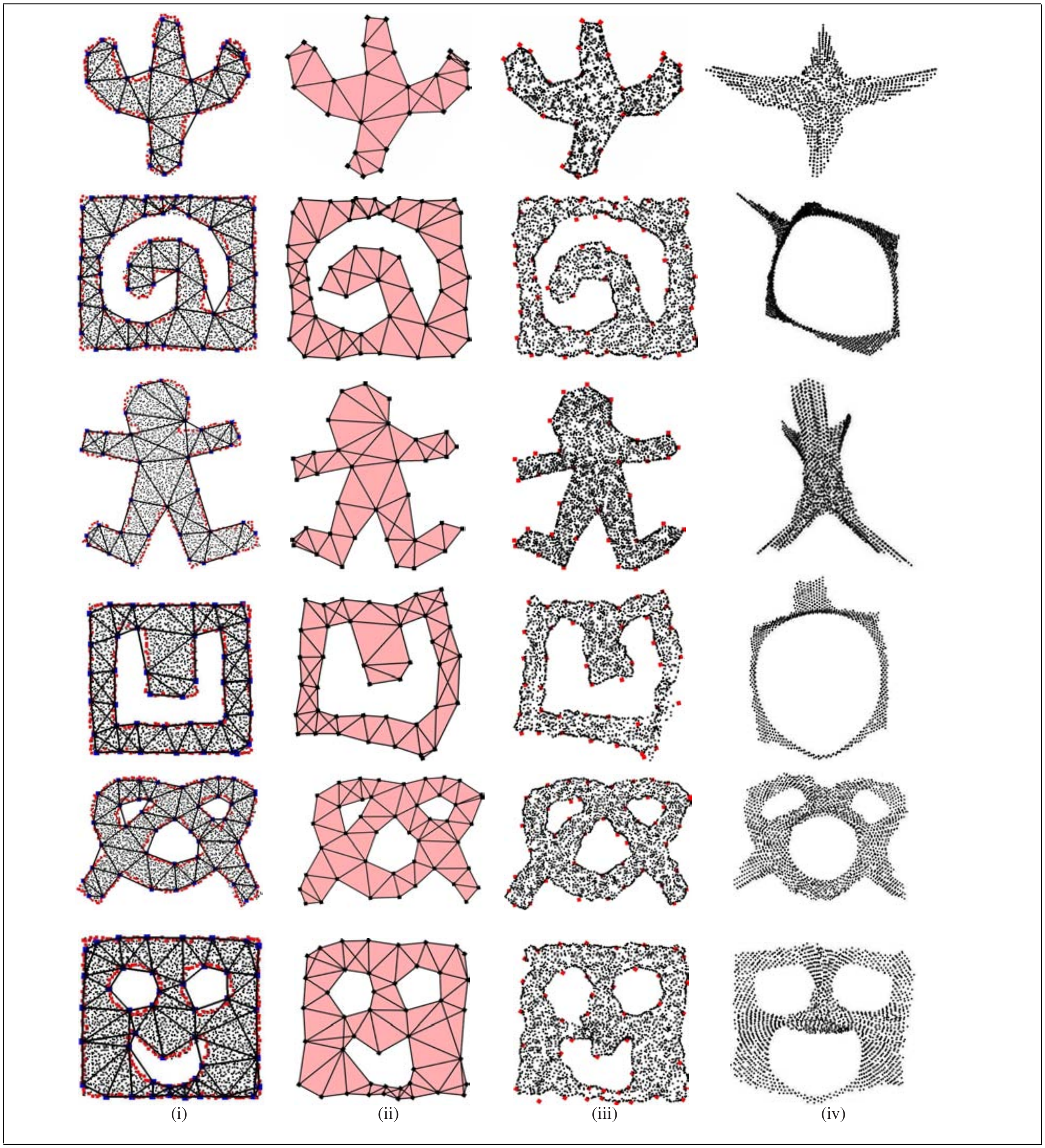


Figure 13: From left to right, we have: (i) the true sensor locations and extracted combinatorial Delaunay complex; (ii) embedding of the combinatorial Delaunay complex; (iii) localization of all nodes by our algorithm; (iv) the results produced by MDS on all nodes in the network. The connectivity network 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: Spiral in a box, 2910 nodes with average degree of 9.5. Third row: Ginger man, 2807 nodes with average degree of 10. Fourth row: Square with a concave hole, 2161 nodes with average degree of 10.4. Fifth row: Pretzel, 2993 nodes with average degree of 9.1. Sixth row: Smile face, 2782 nodes with average degree of 9.5.

addition, at the heart of MDS is singular value decomposition (SVD) which has a complexity of $O(n^3)$, making MDS sluggish. 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.

Rubberband representation. In rubberband embedding [19, 36], 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. Most of them look similar to Figure 4 (right). Due to space constraint we omit the results for additional examples.

We do use rubberband relaxation in conjunction with our landmark embedding algorithm in embedding non-landmark nodes. Since landmark embedding already recovers the precise shape, both of the outer boundary and of any interior cycles, we apply rubberband relaxation to improve the localization quality for the remaining nodes.

Simulation results. We applied our algorithm to a number of networks with different layouts, or "shapes", and were able to recover the original shape quite faithfully. Figure 13 (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. MDS gives reasonable results for some cases (such as the 5th and 6th 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 2nd and 4th examples.

5 Conclusion

In this paper we proposed an efficient and distributed algorithm to discover and recover the sensor network layout. The protocol is particularly attractive for large-scale sensor deployment with holes and complex shape. The novelty of our scheme is to extract high-order topological information to solve the notoriously difficult problem of resolving flip ambiguities in localization algorithms. While geometric information of sensor nodes (e.g. node locations) has been recognized as an important character in sensor networks, the topology of the sensor field is important as well.

Acknowledgement. This work is supported by NSF CAREER Award CNS-0643687 and funding from Stony Brook's Center of Excellence in Wireless and Information Technology. We thank Alexander Kröller, Joe Mitchell, Rik Sarkar, Xianjin Zhu for various discussions on this problem.

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22(4):481–504, 1999.

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

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

[4] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 46–54, 2004.

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

[6] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of Lipschitz surfaces. In *Proc. 22nd Annual ACM Sympos. Comput. Geom.*, pages 337–346, 2006.

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

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

[9] F. Cazals and J. Giesen. Delaunay triangulation based surface reconstruction. In J.-D. Boissonnat and M. Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*. Springer-Verlag, Mathematics and Visualization, 2006.

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

[11] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, S. Morse, B. Anderson, and P. Belhumeur. Rigidity, computation, and randomization of network localization. In *Proceedings of IEEE INFOCOM*, 2004.

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

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

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

[15] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In *ALGOSENSORS*, volume 3121 of *Lecture Notes in Computer Science*, pages 123–136, 2004.

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

[17] S. Funke. 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*, pages 44–53, 2005.

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

[19] S. Funke and N. Milosavljevic. Network sketching or: “how much geometry hides in connectivity? - part II”. In *18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[20] R. Ghrist and A. Muhammad. Coverage and hole-detection in sensor networks via homology. In *Proc. the 4th International Symposium on Information Processing in Sensor Networks (IPSN’05)*, pages 254–260, 2005.

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

[22] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. Anderson. Network localization in partially localizable networks. In *Proceedings of IEEE INFOCOM*, 2005.

[23] C. Gotsman and Y. Koren. Distributed graph layout for sensor networks. In *Proceedings of the International Symposium on Grpah Drawing*, pages 273–284, September 2004.

[24] B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, 1992.

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

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

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

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

[29] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Engineering Math.*, (4):331–340, 1970.

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

[31] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *IEEE GLOBECOM*, pages 2926–2931, 2001.

[32] D. Niculescu and B. Nath. *Ad hoc* positioning system (APS) using AOA. In *IEEE INFOCOM*, volume 22(1), pages 1734–1743, 2003.

[33] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems (APS). In *MobiHoc ’04: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 20–30, 2004.

[34] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-free distributed localization in sensor networks. Technical Report TR-892, MIT LCS, 2003.

[35] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom ’00: Proceedings of the 6th ACM Annual International Conference on Mobile Computing and Networking*, pages 32–43, 2000.

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

[37] A. Savvides, C.-C. Han, and M. B. Strivastava. 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*, pages 166–179, 2001.

[38] A. Savvides, H. Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *ACM Mobile Networks and Applications (Special Issue on Wireless Sensor Networks and Applications)*, 8(4):443–451, 2003.

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

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

[41] K. Whitehouse and D. Culler. A robustness analysis of multi-hop ranging-based localization approximations. In *IPSN ’06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 317–325, 2006.