Stony Brook University

Geometry Discovery with Connectivity Information and Applications in Sensor Networks

A Dissertation Presented

by

Yue Wang

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

May 2009

Stony Brook University
The Graduate School

Yue Wang

We, the dissertation committee for the above candidate for
the degree of Doctor of Philosophy, hereby recommend
acceptance of this dissertation.

Dr. Joseph S.B. Mitchell - Dissertation Advisor
Professor, Computer Science Department and
Applied Mathematics and Statistics Department

Dr. Jie Gao - Chairperson of Defense
Assistant Professor, Computer Science Department

Dr. Esther M. Arkin - Committee Member
Professor, Computer Science Department and
Applied Mathematics and Statistics Department

Dr. Jiaqiao Hu - Outside Committee Member
Assistant Professor, Applied Mathematics and Statistics Department,
Stony Brook University

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Geometry Discovery with Connectivity Information and Applications in Sensor Networks

by

Yue Wang

Doctor of Philosophy

in

Computer Science

Stony Brook University

2009

Wireless sensor networks are tightly associated with the underlying environment in which the sensors are deployed. The global geometry and topology of the network is of great importance to both sensor network applications and the implementation of networking functionalities. In this thesis, we contribute a comprehensive framework for the discovery of sensor network geometry based only on connectivity information, and apply it to several challenging problems: Boundary Recognition, Homology Computation, Layout Recovery and Localization

We develop a distributed algorithm to detect the nodes on the boundaries in a sensor network by using only connectivity information. Our algorithm is motivated by an observation that holes in a sensor field create irregularities in hop count distance. Inner holes of the sensor field disrupt the natural flow of the shortest path tree. We then exploit special structure of the shortest path tree to detect the existence of holes. We also connect those boundary nodes into meaningful boundary cycles and obtain the medial axis of the sensor field as a byproduct. We show by extensive simulation that our boundary detection algorithm gives good results even for networks with low density. The correctness of the algorithm for a continuous geometric domain bounded by polygonal obstacles is proved rigorously as well.

We introduce a new quantity, called homotopy feature size (hfs), for bounded domains. It measures half the length of the shortest loop through point x in domain X. The resort to an intrinsic metric makes this feature size rather insensitive to the local geometry of the domain. Therefore, hfs can lead to a reduced number of samples which still capture the topology of X. We propose algorithms for estimating the hfs, selecting a landmark set of sufficient density, and computing the homology of domain X using the geodesic witness complex $\mathbb{C}_X^W(L)$ and a relaxed version $\mathbb{C}_{X,\nu}^W(L)$. We also present some practical simulations in the context of sensor networks that corroborate our theoretical results.

We propose a distributed algorithm to discover and recover the layout of a large 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. Moreover, 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 also prove the global rigidity of the combinatorial Delaunay complex in the case of a continuous geometric region.

A limitation of our original layout discovery algorithm is its reliability of boundary detection result. As a follow up to the previous algorithm, we develop a new landmark selection algorithm with incremental Delaunay refinement. The new algorithm does not assume any knowledge of the network boundary and runs in a distributed manner to select landmarks incrementally until both the global rigidity property (the Delaunay complex is globally rigid and thus can be embedded uniquely) and the coverage property (every node is not far from the embedded Delaunay complex) are met. The new algorithm improves the robustness and applicability of the original localization algorithm substantially.

Dedicated to my parents, my brother's family and Xiang.

Contents

List of Tables

List of Figures

Acknowledgements

I am grateful to my advisor, Prof. Joseph S.B. Mitchell for his countless advices and guidance during my graduate studies. Joe opened the door of geometry research to me and he was always there when I had any questions. I really appreciate his persistent support, encouragement and helps, especially during my hardest time to study part-time in the past three years.

I am grateful to Prof. Jie Gao. It's my pleasure to work with Jie closely in the past several years. From Jie, I not only gained deep insights into the research areas, but also got a lot of life and career suggestions, which are very precious for me.

I would like to thank all my coauthors besides Joe and Jie: Sol Lederer, Prof. Leonidas J. Guibas, Dr. Steve Y. Oudot. It was fun and productive to work with them.

I would like to thank Prof. Esther M. Arkin and Prof. Jiaqiao Hu, for being my committee members and giving me feedback on my dissertation work. I also want to thank Prof. Wei Zhu, Prof. Michael Bender, Prof. Steve Skiena, and all other faculties and staffs in Computer Science Department as well as Applied Mathematics and Statistics Department. Their kindness and help are unforgettable things during my time at Stony Brook.

I also want to thank all my friends at Stony Brook for their suggestions, helps and supports.

Finally, I want to thank my parents, my brother's family and my husband Xiang for their endless love for me, confidence in me anytime and anywhere. Nothing would be possible for me without them.

Publications

1. **Yue Wang**, Sol Lederer, Jie Gao, Connectivity-based Sensor Network Localization with Incremental Delaunay Refinement Method, Proc. of the 28th Annual IEEE Conferenceon Computer Communications (INFOCOM'09), April, 2009. Also presented at 18th Fall Workshop on Computational Geometry, Oct 31-Nov 1, 2008.
2. Sol Lederer, **Yue Wang**, Jie Gao, Connectivity-based Localization of Large ScaleSensor Networks with Complex Shape, Proc. of the 27th Annual IEEE Conferenceon Computer Communications (INFOCOM'08), 789-797, May, 2008. Journal version accepted and to appear in ACM Transactions on Sensor Networks, 2008.
3. Jie Gao, Leonidas J. Guibas, Steve Y. Oudot, **Yue Wang**, Geodesic Delaunay Triangulation and Witness Complex in the Plane, Proc. of ACM-SIAM Symposium on Discrete Algorithms(SODA'08), 571-580, January, 2008. Full version invited to the special issue of Transactions on Algorithms on SODA'08.
4. **Yue Wang**, Jie Gao, Joseph S.B. Mitchell. Boundary Recognition in Sensor Networks By Topological Methods. The 12th Annual International Conference on Mobile Computing and Networking (MobiCom'06), September, 2006.
5. Andrew Mehler, Yunfan Bao, Xin Li, **Yue Wang**, Steven Skiena, Spatial Analysis of News Sources, IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2006).

Chapter 1

Introduction

In this chapter, we first give a background review of geometry discovery in sensor network, the core problem of our work. We then summarize our main contributions. At last, we briefly introduce contents of each chapter, and the overall organization of the thesis.

1.1 Background review

1.1.1 Problem statement

Wireless sensor networks are tightly coupled with the geometric environment in which they are deployed. On one hand, sensor network applications such as environment monitoring and data collection require sufficient coverage over the region of interest. On the other hand, the global geometry and topology of a wireless sensor network plays an important role in the design of basic networking functionalities, such as point-to-point routing and data gathering mechanisms.

In this thesis, the first problem we will study is an important property of the global geometry in sensor networks, which is nodes detection on the boundaries (both inner and outer boundaries). The viewpoint we take is to regard the sensor network as a discrete sampling of the underlying geometric environment. This is motivated by the fact that sensor networks are to provide dense monitoring of the underlying space. Thus, the shape of the sensor field, e.g., the boundaries, indicates

important features of the underlying environment. These boundaries usually have physical correspondences, such as a building floor plan, a map of a transportation network, terrain variations, and obstacles (buildings, lakes, etc). Holes may also map to events that are being monitored by the sensor network. If we consider the sensors with readings above a threshold to be "invalid", then the hole boundaries are basically iso-contours of the landscape of the attribute of interest. Examples include the identification of regions with overheated sensors or abnormal chemical contamination. Holes are also important indicators of the general health of a sensor network, such as insufficient coverage and connectivity. The detection of holes reveals groups of destroyed sensors due to physical destruction or power depletion, where additional sensor deployment is needed.

Besides the practical scenario mentioned above, understanding the boundary of the sensor field is of great importance in the design of basic networking operations. For example, in the sensor deployment problem, if we want to spread some mobile sensors in an unknown region formed by static sensor nodes, knowing the boundary of the region allows us to guarantee that newly added sensors are deployed only in the expected region. A number of networking protocols also exploit geometric intuitions for simplicity and scalability, such as geographical greedy forwarding [17, 64]. Such algorithms based on local greedy advances would fail at local minima if the sensor networks have non-trivial topology. Backup methods, such as face routing on a planar subgraph, can help packets get out of local minima but create high traffic on hole boundaries and eventually hurt the network lifetime [17, 64]. This artifact is not surprising, since any algorithm with a strong use of geometry, such as geographical forwarding, should adhere to the genuine shape of the sensor field. Recently, there are a number of routing schemes that address explicitly the importance of topological properties and propose routing with virtual coordinates that are adaptive to the intrinsic geometric features [19, 37]. The construction of these virtual coordinate systems requires the identification of geometric features first.

Besides boundary detection, another interesting question related the global geometry of sensor network is can we capture the topology of space X using as few as possible samples? Specifically, motivated by landmarking strategy and manifold

sampling technique, which are extensively used in geometric data analysis recently, several questions arise naturally: (1) how many landmarks are necessary to capture the invariants of a given object X at a given scale? (2) what data structures should be built on top of them? (3) Can we find a homology of the original space X? Actually, these questions are meaningful to a lot of situations where a topological domain or space X is known to us only through a finite set of samples. Understanding global topological and geometric properties of X through its samples is important in a variety of applications, including surface parametrization in geometry processing, non-linear dimensionality reduction for manifold learning, routing and information discovery in sensor networks, etc. In this thesis, we only focus on understanding topological equivalence of sensor network.

Till now, our emphasis is to identify large topological features such as coverage holes, boundaries, topology equivalence. However, there is still an unclosed gap in the loop, as Funke and Milosavljevic spotted [47], in that the identification of boundary nodes or holes numbers still has not produced a global picture of the sensor field layout. Or, we know which are the boundary nodes and how many holes there are but we have no idea how they are laid out in the domain. This is the most challenging problem we will solve in this thesis: with connectivity information only, can we recover the global layout of the network? Furthermore, can we discover the true node location with the help of the identification of the network geometric features?

The physical location of sensor nodes in a network is critical for network operation and data interpretation. It 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. Unfortunately, as sensor networks scale in size, retrieving the locations of the nodes becomes even more challenging. The difficulty is not only due to the network scale, error accumulation, and the 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 irregular shape 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 1 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.

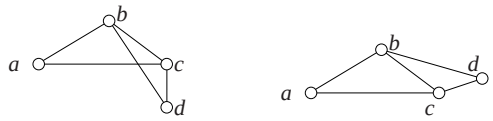


Figure 1: A connectivity graph with two distinct embeddings having the s set of edge lengths.

The network layout is of great help for large-scale network localization and also interesting on its own. For example, greedy routing with imperfect geographical locations that differ from their true locations can still achieve high success delivery rate [80], if there are no global flips. To give an intuitive feeling, Figure 2 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.

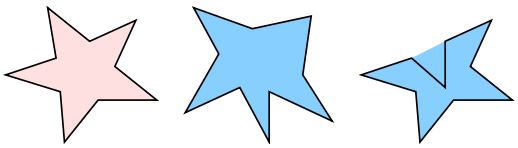


Figure 2: Left to right: the ground truth; one possible embedding; a more devastating embedding.

1.1.2 Prior and related work

1.1.2.1 Boundary recognition

Existing work on boundary recognition can be classified into three categories by their major techniques: geometric methods, statistical methods and topological methods.

Geometric methods for boundary detection use geographical location information. The first paper on this topic, by Fang *et al.* [36], assumes that the nodes know their geographical locations and that the communication graph follows the unit-disk graph assumption, where two nodes are connected by an edge if and only if their distance is at most 1. The definition of holes in [36] is intimately associated with geographical forwarding such that a packet can only get stuck at a node on hole boundaries. Fang also proposed a simple algorithm that greedily sweeps along hole boundaries and eventually discovers boundary cycles.

Statistical methods for boundary detection usually make assumptions about the probability distribution of the sensor deployment. Fekete *et al.* [40] proposed a boundary detection algorithm for sensors (uniformly) randomly deployed inside a geometric region. The main idea is that nodes on the boundaries have much lower average degrees than nodes in the "interior" of the network. Statistical arguments yield an appropriate degree threshold to differentiate boundary nodes. Another statistical approach is to compute the "restricted stress centrality" of a vertex v, which measures the number of shortest paths going through v with a bounded length [39]. Nodes in the interior tend to have a higher centrality than nodes on the boundary. With a sufficiently high density, the centrality of the nodes exhibits bi-modal behavior and thus can be used to detect boundaries. The major weakness of these two algorithms is the unrealistic requirement on sensor distribution and density: the average degree needs to be 100 or higher. In practice the sensors are not as dense and they are not necessarily deployed uniformly randomly.

There are also topological methods to detect insufficient sensor coverage and holes. Ghrist [52] proposed an algorithm that detects holes via homology with no

knowledge of sensor locations; however, the algorithm is centralized, with assumptions that both the sensing range and communication range are disks with radii carefully tuned. Kröller [68] presented a new algorithm by searching for combinatorial structures called flowers and augmented cycles. They make less restrictive assumptions on the problem setup, modeling the communication graph by a quasi-unit disk graph, with nodes p and q definitely connected by an edge if $d(p,q) \leq \sqrt{2}/2$ and not connected if $d(p,q) > 1$. The success of this algorithm critically depends on the identification of at least one flower structure, which may not always be the case especially in a sparse network.

Towards a practical solution, Funke [44] developed a simple heuristic with only connectivity information. The basic idea is to construct iso-contours based on hop count from a root node and identify where the contours are broken. Under the unit-disk graph assumption and sufficient sensor density, the algorithm outputs nodes marked as boundary with certain guarantees. Specifically, for each point on the geometry boundary, the algorithm marks a corresponding sensor node within distance 4.8, and each node marked as boundary is within distance 2.8 from the actual geometry boundary [45]. The simplicity of the algorithm is appealing; however, the algorithm requirement of the algorithm is also rather high; in order to obtain good results, the average degree generally needs to be at least 15.

1.1.2.2 Homology computation

In order to capture the topology of an unknown manifold from a finite set of sampling, several geometric tools or concepts, such as landmark, manifold sampling are the good start. Given a point cloud W sampled from a hidden domain or space X, the idea is to select a subset $L \subset W$ of landmarks, on top of which some data structure is built to encode the geometry and topology of X at a particular scale. Examples in data analysis include the topology estimation algorithm of [30] and the multi-scale reconstruction algorithm of [14, 57]. Both algorithms rely on the structural properties of the *witness complex*, a data structure specifically designed by de Silva [29] for use with the landmarking strategy. Examples in sensor networks include the GLIDER routing scheme and its variants [37, 38]. The idea underlying

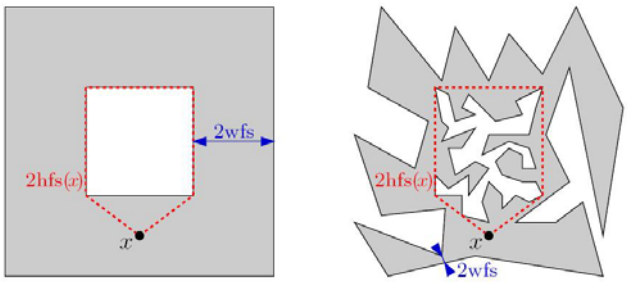


Figure 3: Two Lipschitz domains with very different weak feature sizes (wfs), but similar homotopy feature sizes(hfs).

these techniques is that the use of sparse landmarks at different density levels enables us to reduce the size of the data structures, and to perform calculations on the input data set at different scales.

Manifold sampling issues have been intensively studied in the past, independently of the context of landmarking. The first results in this vein were obtained by Amenta, Bern, and Eppstein, for the case where X is a smoothly-embedded closed curve in the plane or surface in 3-space [2, 3]. Their bound on the landmark density depends on the local distance to the medial axis of $R^2 \setminus X$ (the *local feature size*), and the data structure built on top of L is the so-called *restricted Delaunay triangulation*. Several extensions of their results have been proposed, to deal with noisy data sets [31], sampled from closed manifolds of arbitrary dimensions [14, 26], smoothly or non-smoothly embedded in Euclidean spaces [15]. In parallel, others have focused on unions of congruent Euclidean balls and their topological invariants, which can be computed via the dual complex – known as the Čech complex. In a seminal paper [77], Niyogi *et al.* proved that, if X is a smoothly-embedded closed manifold and L a dense enough sampling of X, then, for a wide range of values of r, the union of the Euclidean open balls of radius r about the points of L deformation retracts onto X.

The above results hold only for closed manifolds. The presence of boundaries (like the scenario in sensor network) brings in some new issues and challenges. An interesting class of manifolds with boundaries is the one of bounded domains in R^n. These naturally arise in the configuration spaces of motion planning problems in robotics, in monitoring complex domains with sensor networks, and in many other contexts where natural obstacles to sampling certain areas exist. By studying

the stability of distance functions to compact sets in R^n, Chazal and others have extended the results of Niyogi *et al.* to a larger class of objects, including all bounded domains X with piecewise-analytic boundaries [24]. Their bound on the landmark density depends on the so-called *weak feature size* of X, defined as the smallest positive critical value of the Euclidean distance to ∂X. This mild sampling condition makes the results of [24] valid in a very general setting. However, in many cases the weak feature size is small compared to the size of the topological features of X, as illustrated in Figure 3 (right). As a result, many sample points are wasted satisfying the sampling condition of [24], when very few could suffice to capture the topology of X. In practice, this results in a considerable waste of memory and computation power.

The case of bounded domains suggests the use of an intrinsic metric on the domain, instead of the extrinsic metric provided by the embedding. This is essential for certain classes of applications, such as sensor networks, where node location information may not be available and only the geodesic distance can be approximated via wireless connectivity graph distances. Intrinsic metrics have been studied in the context of Riemannian manifolds without boundary [71] and, from a more computational point of view, in the context of the so-called *intrinsic* Delaunay triangulations (iDT) of triangulated surfaces without boundary [13]. 2-D triangle meshes in 3-D that happen to coincide with the iDT of their vertices are known to have many attractive properties for PDE discretization [41], and generating such iDT meshes is a topic of considerable interest in geometry processing [32].

1.1.2.3 Layout recovery and localization

For layout and localization problems of sensor network, Many localization algorithms have been proposed in the past few years [35,53–55,74,75,79,82,84,86, 87], yet there is still no universally recognized localization algorithm that produces accurate location information with small overhead [76,91]. As we have pointed out earlier, a major challenge in network localization is to resolve flip ambiguities and the rigidity issue. Two triangles sharing an edge can be embedded in two possible ways, with the two triangles on the same side, or on opposite sides of the common edge. In general, whether a graph has a unique embedding or not is investigated

in graph rigidity theory [56]. A graph is rigid in 2D if a realization of the graph in the plane cannot be continuously deformed without changing the lengths of the edges. A graph is globally rigid if it has a unique embedding in the plane given the edge lengths. Graph rigidity in 2D has been relatively well understood. Both a combinatorial characterization of globally rigid graphs and polynomial algorithms for testing such graphs are known [10, 60]. 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 an NP-complete problem [6, 8, 85]. Even if we know that a graph is globally rigid in 2D, there is no known efficient algorithm to find the realization of the graph with the given edge lengths.

The pioneer work of using rigidity theory in network localization [5, 11, 35, 53, 54, 74, 87] focuses on identifying special graphs that do admit efficient localization algorithms. The first idea is to use trilateration graphs [35, 53, 54, 74]. 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), and thus may require more edges than necessary to uniquely embed the graph. 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 realization 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 [11, 87] have 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 the network has sufficiently many edges to be globally rigid.

An approach on anchor-free localization with only network connectivity is to

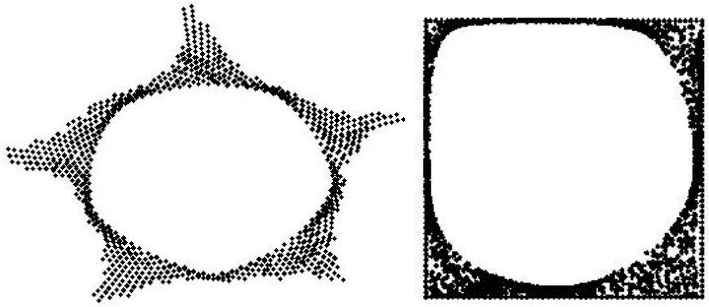


Figure 4: Embedding of the double star. Left: multi-dimensional scaling; Right: rubber-band representation.

use global optimization such as multi-dimensional scaling (MDS) [86]. MDS takes an inter-distance matrix on n nodes and extracts the node location in $\mathbb{R}^n$. For 2D embedding, the locations are taken as the largest 2D linear projection. Figure 4 (left) shows an example. Intuitively, MDS tries to stretch the network out in every direction. For well-connected dense network it gives ok localization result. But it does not have any notion of rigidity and may produce results with global flips. See more examples in Figure 36 of Chapter 4.

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. Actually, our first two problems to be solved in this thesis belong to this category. The focus is to identify high-order topological features (such as holes) from network connectivity [39, 40, 44, 45, 68, 89] and the construction of virtual coordinate systems with which one can route around holes [19, 37, 38, 46, 47]. These virtual coordinates are by no means close to the real node coordinates – they are not meant to be close. Therefore, there is still an interesting question: Can we discover the true node location if we can identify some network geometric features (network boundaries, holes, etc)? One such work is to use the rubberband embedding, by Rao *et al.* in [80] and by Funke and Milosavljevic in [47]. 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 [47, 80] the rubberband representation is mainly used in assigning virtual coordinates to the nodes for geographical routing purposes.

1.2 Our contributions

Our main contributions in this thesis are as follows.

1. We develop a practical distributed algorithm for boundary detection in sensor networks, using only the communication graph (i.e., connectivity information), and not making unrealistic assumptions. We do not assume any location information, angular information or distance information. More importantly, we do not require that the communication graph follows the unit disk graph model or the quasi-unit disk graph model. Our methods also readily provide other topological and geometric information, such as the number of holes (genus), the nearest hole to any given sensor, and the sensor field's medial axis (the collection of nodes with at least two closest boundary nodes) [89].
2. We introduce a new quantity, called the *homotopy feature size*, or hfs for short. hfs depends essentially on the global topology of domain X, and it is rather insensitive to the local geometry. As a result, it enables to have very sparse sets of landmarks L, which makes it a convenient theoretical tool for geometric data analysis. We also prove that the geodesic Delaunay triangulation $D_X(L)$ of L is homotopy equivalent to X. With more practical applications in mind, we have focused on the geodesic witness complex $\mathbb{C}_X^W(L)$ and its relaxed version $\mathbb{C}_{X,\nu}^W(L)$, proving that these two complexes sandwich the geodesic Delaunay triangulation $D_X(L)$ under some conditions. We then estimate the homology of X from L without actually building $D_X(L)$ explicitly, by constructing $\mathbb{C}_X^W(L)$ and $\mathbb{C}_{X,\nu}^W(L)$ and computing their persistent homology [50].
3. We propose an efficient and distributed algorithm to discover and recover the sensor network layout. 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. We take samples on the network boundaries with sufficient density and denote select nodes as landmarks. We then construct the Voronoi diagram and its dual combinatorial Delaunay complex on these landmarks. We prove in the case of a continuous geometric domain that when the landmarks are sufficiently dense, the induced Delaunay graph is rigid and there is a unique way to embed these 'solid' Delaunay triangles in the plane. With the landmarks localized and the network layout successfully recovered, the landmarks serve as 'anchor' nodes such that each additional node can localize itself by using trilateration [70].

4. As a follow-up of [70], in which the dependency on the boundary detection algorithm puts limitations on the applicability of the localization algorithm, we propose an incremental landmark selection algorithm that does not assume knowledge of the network boundary. In particular, we start with no knowledge of the network topology (whether there are holes or how many there are, etc.) and develop local conditions to test whether a node should be included as a new landmark. The landmarks selected naturally adapt to the local geometry of the network, with a higher density of landmark nodes selected in regions with more detailed and complex features. This new algorithm greatly enhances the robustness of our algorithm in cases of extremely sparse or even non-rigid networks, or networks with very complicated shapes that are challenging for boundary detection algorithms [90].

1.3 Thesis organization

This thesis is organized in the following fashion. In Chapter 2, we begin with an introduction on our boundary recognition problem. We prove the correctness of our algorithm for a continuous geometric domain bounded by polygonal obstacles. Extensive simulation is also followed which shows our algorithm indeed gives good results even for networks with low density. Chapter 3 describes the homotopy feature size, geodesic Delaunay triangulation, witness complex and the main properties of these concepts. We give our algorithms for sampling Lipschitz domains

in the plane, estimating their homotopy feature size, and computing their homology. These algorithms are adapted to the sensor networks setting. In Chapter 4, we present our layout discovery and localization algorithm. We prove the rigidity of the Delaunay complex when landmarks are sufficiently dense in the case of a continuous domain. Corresponding simulation results are presented as well. Chapter 5 is a follow up work to Chapter 4. We develop a new landmark selection algorithm with incremental Delaunay refinement. The new algorithm does not assume any knowledge of the network boundary which greatly enhances the robustness of the algorithm. Finally we summarize our work, and give future research directions in Chapter 6.

Chapter 2

Boundary Recognition in Sensor Networks By Topological Methods

2.1 Introduction

In this chapter, we introduce our boundary recognition algorithm in sensor network. Our method only requires connectivity information. It does not assume any knowledge of the node locations or inter-distances, nor does it enforce that the communication graph follows the unit disk graph model. Indeed, actual communication ranges are not circular disks and are often quite irregularly shaped [49]. Algorithms that rely on the unit disk graph model fail in practice (e.g., the extraction of a planar subgraph by the relative neighborhood graph or Gabriel graph [65]). While the unit (or quasi-unit) disk graph assumption is often useful for theoretical analysis, it is preferable to consider algorithms that do not rely on this assumption or that degrade gracefully as the ground truth deviates only modestly from the model. Therefore, we do not put a hard restriction on the communication model in our algorithm. Rather, we use a loose notion of locality in wireless communications: Nearby nodes can communicate directly, and faraway nodes generally do not.

Our boundary detection algorithm is motivated by an observation that holes in a sensor field create irregularities in hop count distances. Simply, in a shortest path tree rooted at one node, each hole is "hugged" by the paths in a shortest path tree. We identify the "cut", the set of nodes where shortest paths of distinct homotopy

types[1] terminate and touch each other, trapping the holes between them. The nodes in a cut can be easily identified, since they have the property that their common ancestor in the shortest path tree is fairly far away, at the other side of the hole. The detection of nodes in a cut can be performed independently and locally at each pair of adjacent nodes.

When there are multiple holes in the network (indicated by multiple branches of the cut), we can explicitly remove all of the nodes on cut branches except one, thereby connecting multiple holes into one. Our algorithm then focuses on finding the inner and outer boundaries of the network, which, with the cut nodes put back, will give the correct boundary cycles. In a network with only one hole (and one cut branch), one can easily find a hole-enclosing cycle. Indeed, for a pair of nodes that are neighbors across a cut (a "cut pair"), the concatenation of the paths from each node in a cut pair to their common ancestor gives such a cycle. This "coarse" boundary cycle is then refined to bound tightly both the inner boundary and outer boundary. In addition to discovering boundary nodes, we also obtain their relative position information, so that we can connect them into a meaningful boundary.

We show by simulation that our algorithm correctly identifies meaningful boundaries for networks with reasonable node density (average degree 6 and above) and distribution (e.g., uniform). The algorithm also works well for non-uniform distributions. The algorithm is efficient. The entire procedure involves only three network flooding procedures and greedy shrinkage of paths or cycles. Further, as a theoretical guarantee, we prove that for a continuous geometric space bounded by polygonal obstacles – the case in which node density approaches infinity – the algorithm correctly finds all of the boundaries.

2.2 Topological boundary recognition

Suppose a large number of sensor nodes are scattered in a geometric region with nearby nodes communicating with each other directly. Our goal is to discover

[1]Two paths have distinct homotopy types if one cannot be "continuously deformed" into the other.

the nodes on the boundary of the sensor field, using only local connectivity information. We propose a distributed algorithm that identifies boundary cycles for the sensor field.

The basic idea is to exploit special structure of the shortest path tree to detect the existence of holes. Intuitively, inner holes of the sensor field "disrupt" the natural flow of the shortest path tree: Shortest paths diverge prior to a hole and then meet after the hole. This phenomenon is also understood in terms of the continuous limit, when the shortest path tree becomes the "shortest path map", which we discuss in Section 2.3. We first outline the algorithm and then explain each step in detail.

1. Flood the network from an arbitrary node, r. Each node records the minimum hop count to r. This implicitly builds a shortest path tree rooted at r. We generally prefer to select r as a node on the outer boundary of the sensor field.
2. Determine the nodes that form the *cut*, where the shortest paths of distinct homotopy type meet after passing around holes. Informally, the nodes of a branch of the cut have their least common ancestor (LCA) relatively far away and their paths to the LCA well separated. See Figure 5(ii-iv). If there are multiple branches of the cut, corresponding to multiple holes, delete nodes on branches of the cut in order to merge holes, until there is only one composite hole left in the sensor field.
3. Determine a shortest cycle, R, enclosing the composite hole; R serves as a coarse inner boundary. See Figure 5(v).
4. Flood the network from the cycle R. Each node in the network records its minimum hop count to R. See Figure 5(vi).
5. Detect "extremal nodes" whose hop counts to R are locally maximal. See Figure 5(vii).
6. Refine the coarse inner boundary R to provide tight inner and outer boundaries. These boundaries are in fact cycles of shortest paths connecting adjacent extremal nodes. See Figure 5(viii).
7. Undelete the nodes of the removed cut branches and restore the real inner boundary locally.

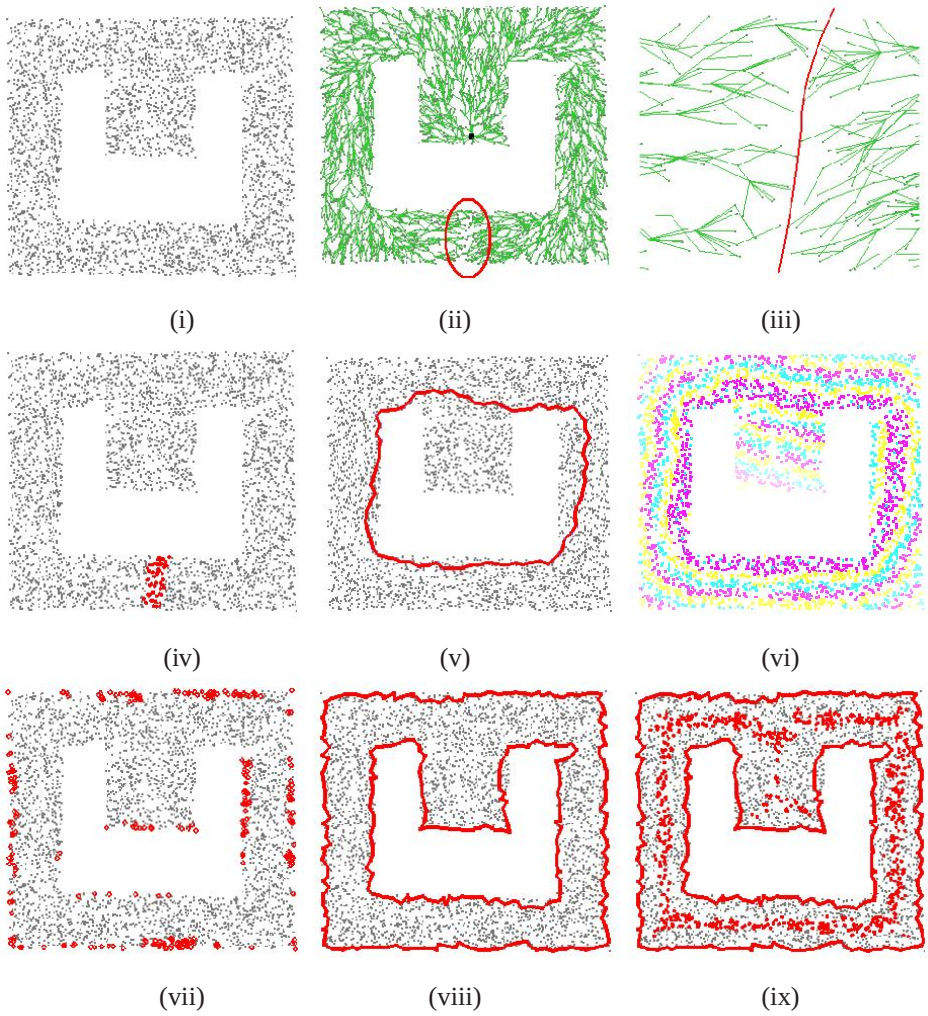


Figure 5: Boundary detection algorithm for an example with one concave hole. The average degree is about 20. (i) The sensor nodes; (ii) The shortest path tree T (green), rooted at the black node; (iii) The zoomed-in portion (within the loop of (ii)) of the tree where the shortest paths meet; (iv) The marked cut nodes (red); (v) The course inner boundary formed by shortest paths from a pair of cut nodes to their least common ancestor; (vi) The nodes colored by hop count (giving iso-contours) in a flooding from the coarse inner boundary; (vii) The nodes (red) with locally maximum hop count (extremal points) from the coarse boundary; (viii) The refinement of the coarse inner boundary, giving tight inner and outer boundaries; (ix) The medial axis nodes of the sensor field, obtained as a by-product of boundary detection.

8. At this stage we have a set of cycles corresponding to the boundaries of the inner holes and the outer boundary. As a byproduct, we can compute the medial axis of the sensor field. See Figure 5(ix).

2.2.1 Build a shortest path tree

The first step of the algorithm is to flood the network from an arbitrary root node r. For example, we can select r to be the node with smallest ID. This can be performed in a distributed fashion as follows. First, each sensor node p sets a timer with a random remaining time. Ties are broken by the unique IDs of the sensor nodes. When the timer of p reaches 0, the node p will begin to flood the network and build a shortest path tree $T(p)$. The message sent out by p contains the ID of p and the initial timer p chooses. The tree with a timer of minimum value starts first and suppresses the construction of other trees. When the frontier of $T(p)$ encounters a node q, there are two cases: (i) If q does not belong to any other tree, then q is included in the tree $T(p)$ and q will broadcast the packet; or, (ii) If q is already included in another tree $T(p')$, then the start time of $T(p)$ and $T(p')$ are compared. Only when the start time of $T(p)$ is earlier than that of $T(p')$, the message from p will be broadcast. Eventually the tree with earliest starting time will dominate and suppress the construction of other trees, and the packet from the node with minimum initial timer will cover the whole network. Each node in the network records the minimum hop count from this root. We denote by T the shortest path tree constructed.

When the network size is unknown, the flooding step provides a good approximation to the network diameter. By triangle inequality, the diameter of the network is at most $2d$, where d is the distance between the root r and the deepest node in T. d can be used to generate a reasonable threshold for the minimum size of the holes we aim to discover.

2.2.2 Find cuts in the shortest path tree

Hints about the presence of holes are hidden in the structure of the shortest path tree T. The "flow" of T forks near a hole, continues along opposite sides of

the hole and then meets again past the hole. We detect where the shortest paths meet and denote those nodes as "cut" nodes (e.g., the red nodes in Figure 5(iv)). Nodes of the cut form *cut branches* and *cut vertices*, where three or more cut branches come together; cut branches are the discrete analogue of bisectors in the (continuous) "shortest path map" and cut vertices are the discrete "SPM-vertices", see Section 2.3 for these concepts in detail. Figure 7 shows the tree T for a network with 3 holes. In this case, we detect 3 groups of cut nodes (3 cut branches). In general, as will be proved later in the continuous case, if the sensor field has k interior holes and m cut vertices, there are exactly $k+m$ cut branches in the network.

Intuitively, the nodes in a cut are the neighboring pairs, (p,q), in the communication graph whose least common ancestor, LCA(p,q), in the shortest path tree T is "far" from p and q, with paths from LCA(p,q) to p and to q "well separated." More formally,

Definition 1 *A* cut pair (p,q) *is a pair of neighboring nodes in the network satisfying the following conditions: (i) The (hop) distance between p or q and y =LCA(p,q) is above a threshold δ_1; and, (ii) The maximum (hop) distance between a node on the path in T from p to y and the path in T from q to y is above a threshold δ_2 (See Figure 6). The* cut *is the union of nodes that belong to a cut pair. Each connected component of the cut is a subgraph which, when thinned, is a* cut tree*; the nodes of degree greater than 2 in this tree are* cut vertices*, and the paths of degree-2 vertices, and their neighboring (non-cut-vertex) cut nodes, form the* cut branches.

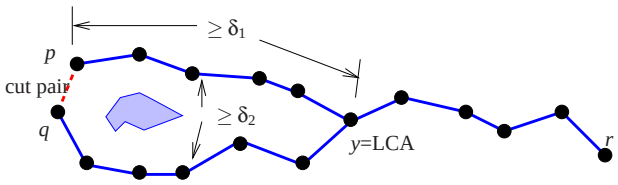


Figure 6: Definition of a cut pair (p,q).

The two parameters in the definition of a cut pair, δ_1 and δ_2, specify the minimum size of the holes we want to detect. Typically, we choose them as some constant fraction of the diameter of the sensor field (e.g., experiments reported in

this thesis use $\delta_1 = \delta_2 = 0.18d$). The condition on whether a node belongs to a cut can be checked locally. Alstrup *et al.* gave a distributed algorithm to compute the LCA [1]. The idea is to label the nodes of a rooted tree with $O(\log n)$ bits such that by using only the labels of two nodes one can calculate the label of their LCA. With this labeling, each pair of neighbors in the network check for their common ancestors. If the LCA is more than distance δ_1 away, we check if two shortest paths from these two neighbor nodes to their LCA satisfy the second condition in the definition above. This can be implemented by a local flooding from each node in the cut pair up to distance δ_2. Nodes that satisfy both conditions mark themselves as being in the cut.

The nodes in the cut will then connect themselves into connected components. Furthermore, each connected component agrees on the node closest to the root (ties are broken by the smaller ID). Specifically, each node u in the cut keeps its current knowledge of the ID of the node with smallest distance to the root. If a cut node u receives a message from its neighboring cut node about a closer node, u updates its knowledge and sends the change to its neighboring cut nodes. Eventually, the cut nodes connect themselves into connected components with each cut identified by the ID of the closest node. If there is no hole in the network, there will be no cut, then no connected components correspondingly. By simulation we find that this algorithm correctly finds all the cuts when the sensor field has a reasonable density (e.g., when the average degree is about 7 or more).

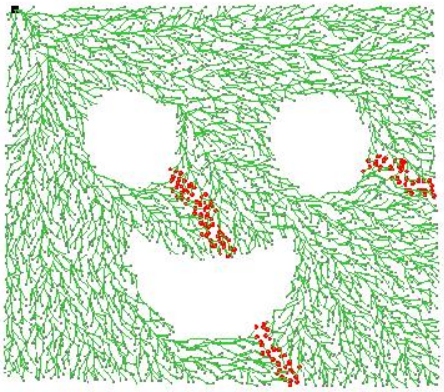


Figure 7: The cut pairs in a multi-hole example. 4050 nodes and the average degree is 10.

When there are multiple holes and multiple cuts in the network, we artificially merge the holes by removing nodes on cut branches, until there is only one composite hole left. As will be clear later, the real boundary cycles can be easily restored by undeleting the removed cut nodes. We remove all of the nodes on cut branches except the one branch furthest away from the root. The interior holes either connect to themselves or connect to the outer boundary. That is, the multi-hole case is turned into a single hole scenario. Thus, in the following steps, we focus on the single hole case.

2.2.3 Detect a coarse inner boundary

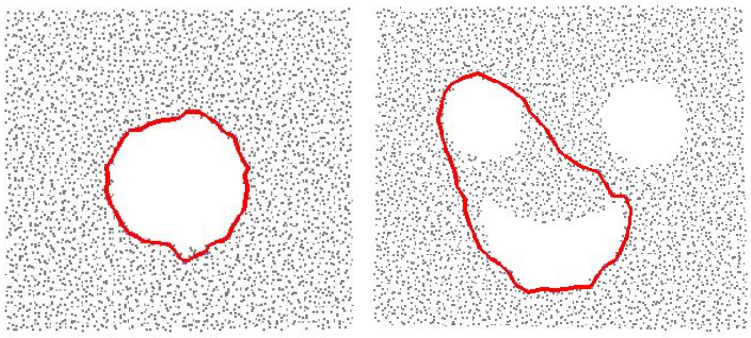
With the cut nodes detected, we would like to find a coarse inner boundary R that encloses the (composite) interior hole. R is then refined to provide tight inner and outer boundaries.

Definition 2 *A* coarse inner boundary *R is a shortest cycle enclosing the interior hole in the sensor field.*

Recall that we have turned any multi-hole sensor field into a single composite hole sensor field by removing all cut branches except one. All the nodes in the unique cut branch have the ID of the node closest to the root. Thus, the closest node p, together with its partner q in the cut pair, will find the shortest paths between them that do not go through any cut node. This can be implemented in two ways. An obvious way is to use any shortest path algorithm to find this path. In order to prevent the path from going through any cut nodes, we remove all the edges between cut pairs. Alternatively, we can use the two shortest paths from p and q to LCA(p,q). Together with the edge pq, we obtain a cycle that encloses the hole. This cycle is not necessarily the shortest cycle. But we can greedily shrink it to be as tight as possible, by the following k-hop shrinking process. For any two nodes that are within k hops on the cycle, we check whether there exists a shorter path between these two nodes. If so, we use the shorter path to replace the original segment on the cycle and shorten the total length. For example, a 2-hop shrinking check, for every three adjacent nodes on the cycle, say x,y,z, whether (x,z) is an

edge. If so, we shrink the cycle by excluding y. In a sensor field with reasonable density, the greedy process stops at the shortest cycle so that no more improvement is made. For a sensor field with only one convex shape hole, the coarse inner boundary actually is the real inner boundary, as in Figure 8(i). If the graph has a concave hole (Figure 5(v)), the coarse inner boundary R will be a shortest circuit containing this concave hole and, thus, approximates its convex hull. A multi-hole example is shown in Figure 8(ii). Notice that this coarse inner boundary provides a consistent ordering of the nodes on this cycle.

We note that the shortest paths from a cut pair to their LCA do not go through any cut nodes, as proved later (Section 2.3). Thus, the removal of cut nodes in a multi-hole scenario is not a problem for the discovery of the coarse inner boundary by the greedy shrinking scheme.



(i) (ii)

Figure 8: The coarse inner boundary for (i) a convex hole scenario; (ii) a multi-hole scenario with all but one cut branch removed, one interior hole connected to the outer boundary.

2.2.4 Find extremal nodes

The coarse inner boundary R is not tight in the case of a concave hole. Also, we have no idea about the outer boundary. Next, we refine the coarse inner boundary to provide tight cycles for both inner and outer boundaries. This refinement is through finding what are called “extremal nodes” with respect to R.

Definition 3 *An* extremal node *is a node whose minimum hop count to nodes in R is locally maximal*

To discover extremal nodes, we have the nodes on R synchronize among themselves and start to flood the network at roughly the same time [34] [48]. Each node in the network records the minimum hop count to nodes in the coarse inner boundary R. This is as if we merge the nodes in R to a dummy root σ, and build a shortest path tree $T(\sigma)$, rooted at σ for the whole network. The extremal nodes are the ones with locally maximum distance to R. Each extremal node can detect itself by checking its direct neighbors. Intuitively, the extremal nodes are on the outer boundary or are the ridges on the real inner boundary of a concave hole (Figure 5(vii)).

With the extremal nodes identified, we also need to know their relative orderings to connect them to a consistent boundary. This ordering can be derived from the ordering of the nodes on R. Specifically, for each node on R, we assign a label in $[0, L]$ that indicates its position on R. A node u that is k hops away from R may have multiple neighbors $(k-1)$ hops away from R. Thus, the label of u, $\ell(u)$, is taken as the average of the labels of all of its neighbors that are $(k-1)$ hops away from R. Each node in the network records the distance to R as well as its label. The relative ordering of extremal nodes is decided by their labels.

When there are many extremal nodes and some of them have similar labels, we would like to eliminate some of them and take a few representative nodes for the next step. The goal for this clean-up is to take sampled extremal nodes so that we can easily sort them. We first find the connected components for all extremal nodes, denoted as extremal connected components (ECC). These can be done with the same distributed strategy as used to find connected components for the cuts. Next, we select at most two nodes as the representatives for each ECC (an ECC may contain only one node). These two nodes have the greatest distance between them in their affiliated ECC. These nodes will be connected according to their labels to refine inner and outer boundaries.

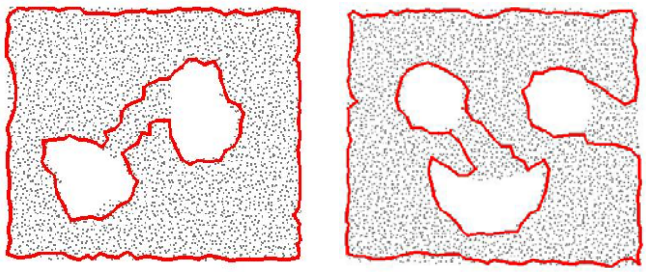
2.2.5 Find the outer boundary and refine the coarse inner boundary

With the extremal nodes and a linear ordering of them, we would like to connect them into a cycle consistent with this ordering. Notice that here we only consider the case with one inner hole, since we have removed the cuts to connect multiple holes into a composite hole. If this composite hole is convex, then all extremal nodes belong to the outer boundary. If the composite hole is concave, as in the multi-hole case, there will be two types of extremal nodes, those on the outer boundary and those on the interior of R. We use the following rule to differentiate extremal nodes on different sides of R.

The removal of R, together with all of the 1-hop neighbors of R, partitions the network into several connected components C_i, $i = 1, \ldots, w$. Now we would like to connect the extremal points into extremal paths in each connected component, which will then be further connected into boundary cycles. We first focus on a particular connected component and describe how an extremal path is constructed. Specifically, all of the nodes that are 2 hops away from R will connect themselves through local flooding. This results in a path (or a cycle) P_j with a corresponding maximum index and minimum index along R. Now we would like to refine this path P_j so as to force it to go through the extremal nodes in this connected component as well as the two nodes on P_j with maximum and minimum indexing. Notice that each extremal node has shortest paths from some nodes on R. Now we connect two extremal nodes with adjacent ordering by the shortest path between them. To find the shortest path between two extremal points, again we can either use any shortest path algorithm or a greedy approach. For the latter, we notice that there is a natural path between any two extremal nodes u, v, composed by the shortest path from u to its closest point on P_j, the shortest path from v to its closest point on P_j and a segment of P_j between the correspondences of u, v. The shortest paths from extremal points to P_j can be found by following the parent pointer towards R (in the tree $T(\sigma)$). Then, we can greedily refine this path and find the shortest path between u and v.

By the above procedure, the extremal nodes are connected into a path P_j, in

each connected component. If one of the paths is actually a cycle, then it is already a boundary cycle. If there are paths that are not connected into cycles, then we tour the coarse boundary R and connect them into boundary cycles. In particular, we start from anywhere on R and tour along R; when there is a neighboring node (in 2 hops) who is on an extremal path P_j, then we branch on P_j. This will eventually come back and close a cycle. Then, we tour R again, from a node that is not on the first cycle, and branch on extremal paths as long as we can. This will generate another cycle. So far, we classify extremal nodes and connect them into two cycles, corresponding to the inner and outer boundary. Figure 9 shows the results for two examples after this stage.



(i) (ii)

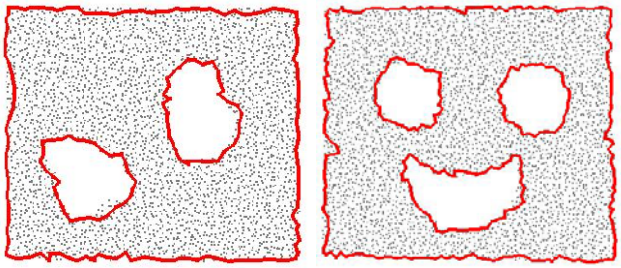
Figure 9: The outer boundary and the refined inner boundary. (i) 2-hole example; (ii) multi-hole example.

As will be shown in Section 2.3, in the continuous case all of the boundary points are identified as extremal points. However, in a discrete network, the shortest path is computed on a combinatorial graph. Thus, not all of the boundary nodes are identified as extremal nodes. The boundary refinement can be performed in an iterative fashion such that we flood from the current boundary cycle and identify more extremal points until the boundary cycle is sufficiently tight.

2.2.6 Restore the inner boundary

The final step of our method is to recover the inner holes in the sensor field and find their boundaries. We undelete the cut nodes we removed earlier and restore the correct inner boundary. For each cut, we find a cut pair (p,q) such that the inclusion

of edge pq in the refined inner boundary R partitions the inner boundary into two boundary cycles. If p, q are by themselves not on the refined inner boundary R, then we do a local search from p, q to discover nodes on R. The two new boundary cycles will share nodes p, q. Then, we shrink the cycles locally to make them tight. Here, the shrinking procedure has a restriction that the shrunk path still has to pass through the extremal nodes. Thus, we partition the refined inner boundary into a number of cycles, each representing the boundary of an inner hole. Figure 10 shows our final results for two examples.



(i) (ii)

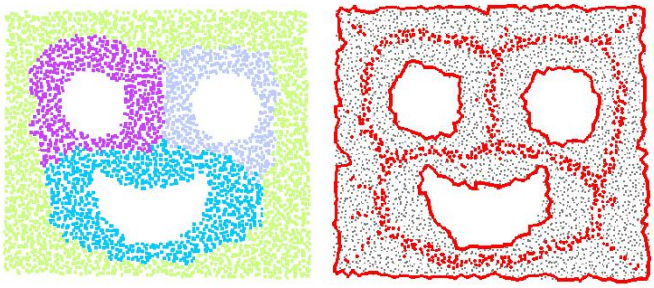
Figure 10: The boundary cycles created by the connection of nodes: (i) 2-hole example; (ii) multi-hole example.

2.2.7 The medial axis of the sensor field

The boundaries of the sensor field capture the global geometric shape information, and thus can be used to generate other structures related to the global geometry. For example, we can define the quasi-Voronoi diagram for the boundaries such that sensor nodes are classified by their closest boundary. In another example, we can compute the medial axis of the sensor field. The medial axis is defined as the set of nodes with at least two closest boundary nodes[2]. The medial axis can be used to generate virtual coordinates for efficient greedy routing [19]. Both the quasi-Voronoi diagram and the medial axis can be discovered by a simple flooding from the boundary cycles. Figure 11 shows the results for our 3-hole face example.

[2]Since hop counts are discrete, we allow a node on the medial axis with two closest boundary nodes that differ by 1 in hop count.

With our boundary information, we start from all boundary nodes and flood the network simultaneously. Each node thus records its closest boundary. All of the nodes with the same closest boundary are naturally classified to be in the same cell of the quasi-Voronoi diagram. Further, the nodes at which the frontiers collide belong to the medial axis. Specifically, the detection of nodes on the medial axis can be done locally. Denote the boundary cycles by R_i, $i = 1, \ldots, k$. During the flooding from the boundaries, each node in the boundary sends out a packet containing the boundary containing it, its position based on the originator boundary, and the number of hops this packet has traveled, as denoted by a tuple $(i, \ell_i(v), h_i(v))$. A node, upon receiving a packet $(i, \ell_i(v), h_i(v))$, does one of the following things. If a node has not received earlier packets with smaller hop counts to boundaries, it records $(i, \ell_i(v), h_i(v))$, where $h_i(v)$ is the minimum hop count distance to the originator of the packet, drops earlier tuples with larger hop counts, and re-transmits the packet. In the end, each node only contains the tuples with the minimum hop count distance. There are three possible cases: (1). Only one minimum hop count tuple is left for the node, so the node is not a medial axis node. (2). More than one minimum hop count tuple is left, and these tuples are generated by different boundary cycles, which means the node has more than one closest boundary node, so the node is on the medial axis. (3). More than one minimum hop count tuple is left, and these tuples are generated by the same boundary cycle, so we need to compare the ℓ values in these tuples; if they differ a lot, then the node is on the medial axis. This case corresponds to an inward convex (reflex) segment in the boundary.



(i) (ii)

Figure 11: (i) A quasi-Voronoi diagram; (ii) the medial axis diagram.

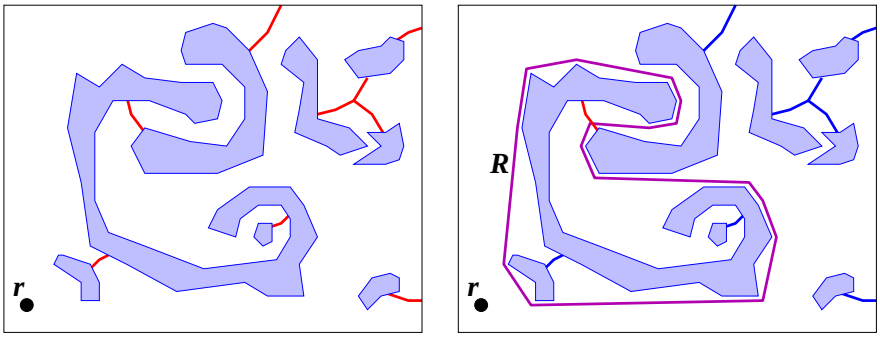
2.3 Proof of correctness in the continuous case

In this section we prove rigorously that the algorithm will correctly detect all the inner and outer boundaries in a continuous domain with polygonal boundaries. (The results apply also to non-polygonal domains, since they are approximated by polygons.) Let F be a closed polygonal domain in the plane, with k simple polygonal obstacles inside; F is a simple polygon P, minus k disjoint polygonal holes. We refer to F as the free space and its complement, O, as the obstacle space. O consists of k open, bounded simple polygonal holes and an unbounded complement of the simple polygon P. Denote by V the set of all vertices of F.

For any two points $p,q \in F$, we denote by $g(p,q)$ the geodesic shortest path in F between p,q. The Euclidean length of $g(p,q)$ is denoted by $d(p,q)$. The shortest path $g(p,q)$ is not necessarily unique. In fact, our algorithm aims to detect points with one or more shortest paths to the root. Rigorously, given a root $r \in F$, the shortest path tree at r is the collection of shortest paths from each point in F to r. A geodesic shortest path is a polygonal path with turn points at vertices of F [28]. We say that a vertex $s \in V$ is a *parent* of a point $p \in F$ if for some geodesic shortest path $g(r,p)$, s is the last vertex along the path $g(r,p) \setminus \{p\}$ at which $g(r,p)$ turns. If the geodesic path is a straight line from r to p, then r is the parent of p. The free space F can be partitioned to maximal regions, called *cells*, such that all the points in the same cell have the same parent or set of parents. This partition is called the *shortest path map*, SPM(r) (see Figure 12). We define the *bisector*, $C(v_i,v_j)$, of two vertices v_i,v_j as the set of points in F with the set of parents $\{v_i,v_j\}$. A point in F with three or more parents is called an *SPM-vertex*. The union of all bisectors and SPM-vertices, B, is often called the *cut locus*. Our boundary detection algorithm makes use of the properties of the shortest path map. We list below the useful properties that are proved in [72].

Lemma 4 ([72]) *Given a closed polygonal region F in the plane, with k simple polygonal obstacles inside. The shortest path map at an arbitrary root $r \in F$ has the following properties.*

1. *Each bisector is the union of a finite set of closed subarcs of a common hyperbola (a straight line is a degenerate case of a hyperbola).*



(i) (ii)

Figure 12: (i) A shortest path map SPM(r) for F with $k = 8$ obstacles (blue), showing the bisector set B (red), which has one SPM-vertex, 7 connected components, and 9 arcs. (ii) The shortest path map SPM(r), and R is the shortest cycle corresponding to the red bisector arc. Other bisector arcs are blue, having become part of the obstacle set for F'.

2. *The collection of bisectors and SPM-vertices, denoted by B, forms a forest.*
3. *There is at least one bisector point on the boundary of each obstacle.*
4. *$F \setminus B$ is simply connected.*

Lemma 5 *For a region F with k polygonal holes inside and m SPM-vertices, there are exactly $k+m$ bisector arcs.*

Proof. This follows from the fact that a tree of v vertices has $v-1$ edges, where, here, $v = k+1+m$ since the bisector arcs connect the k obstacle boundaries, the m SPM-vertices, and the 1 outer boundary. □

In fact, our boundary detection algorithm finds the bisector arcs. For completeness, we re-state the outline of the boundary detection algorithm for the continuous case. The focus is on the correctness of the algorithm so we explain it in a centralized setting. The boundary detection algorithm for F works as follows.

1. Find the bisectors and SPM-vertices; each of them has at least two geodesic shortest paths to the root r.
2. Delete bisector arcs until there is only one bisector arc left. Correspondingly, we connect $k-1$ holes into one composite hole. This results in a new domain F' with only one interior hole O' and one bisector arc.

3. Find the shortest cycle R in F' enclosing the inner hole O'.
4. Refine R to find both the inner boundary and outer boundary of F'. In particular, we use the following algorithm for boundary refinement.
    1. Compute the Voronoi diagram of R in F'. Find the *extremal points*, defined as the points that do not stay on the interior of any shortest paths from points of F' to its closest point on R, denoted by E. Furthermore, we find for each extremal point the closest point(s) on R.
    2. Order the extremal points by the sequence of their closest point(s) on R. The extremal points are naturally connected into paths or cycles. Tour the coarse boundary R and replace the segments of R by their corresponding extremal paths. Touring the boundary twice will come up with two cycles that correspond to the inner and outer boundaries.
3. Undelete the removed bisector arcs. Restore the boundaries of interior holes and the outer boundary.

Next we will prove that this algorithm correctly finds all the boundaries of F. The proof consists of a number of lemmas.

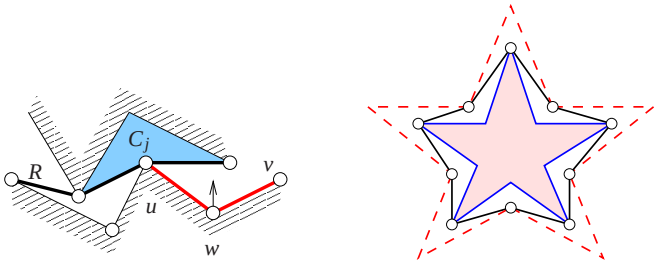
Lemma 6 *The domain F' has one interior polygonal hole.*

Proof. This is due to the fact in Lemma 4 that $F \setminus B$ is simply connected. Thus, by removing all but one bisector arc and all SPM-vertices, we obtain a polygonal region F' with one interior hole. □

Now we focus on F' and argue that we indeed find the correct outer and inner boundaries of F' by the iterative refinement. We only prove for the outer boundary R^+; the correctness of the inner boundary R^- can be proved in the same way. By the refinement algorithm, we can find the Voronoi diagram of R by a wavefront propagation algorithm (e.g., [59]). The extremal points are the points that do not stay on the interior of any shortest paths from points of F' to its closest point on R. Due to the properties of wavefront propagation, the extremal points must be either on the boundary or on the medial axis, where wavefronts collide [61, 72]. However, since cycle R is a shortest path cycle surrounding the hole, we argue that the extremal points on the exterior of R must stay on the boundary of F'.

Lemma 7 *The extremal points are exactly the boundary points of F'.*

Proof. First we argue that all the inward concave vertices of R must be on the outer boundary of F'. By the properties of geodesic shortest paths [73], all the vertices of R must be vertices of F'. If an inward concave vertex is a vertex of the inner hole, then we can move the concave vertex outward and shrink the cycle R, as shown in Figure 13(i) (the vertex of the inner hole w can not be on R). This is a contradiction.



(i) (ii)

Figure 13: (i) All the inward concave vertices of R must be on the outer boundary of F'. (ii) The dark cycle is the coarse inner boundary R. The cycles in blue and (dashed) red are inner and outer boundaries.

Thus, the removal of cycle R partitions the space F' into disjoint components $\{C_j\}$. Correspondingly, we denote by R_j the segment of R that bounds the component C_j. By the above argument, each boundary segment R_j is inward concave. The wavefront propagation of R is actually composed of Voronoi wavefront propagation of each segment of R_j to its bounded region C_j. Since each segment is concave and the C_j's are disjoint, the wavefront propagation does not collide. All the extremal points can only happen at wavefront-boundary collision. In fact, all the boundary points collide with the wavefront propagation and thus are considered as extremal points. □

At this point we can consider R to be a coarse approximation to the outer (and inner) boundary. By the refinement algorithm we will improve the approximation and obtain the correct outer boundary. Specifically, the removal of R from F' partitions the space F' into disjoint components $\{C_j\}$, $j = 1, \ldots, w$. For each connected

component C_j, the extremal points connect themselves into either a path (with open endpoints) or a cycle P_j.

- If $w = 1$, the extremal points form a cycle. R is already the inner boundary. All the extremal points are on the outer boundary.
- If $w \geq 2$ and one of P_j is a cycle, this cycle is the outer boundary. The rest of the extremal points refine R to the inner boundary. Specifically, we replace the segments on R by their extremal paths.
- Otherwise, we will tour R and discover the inner and outer boundaries. Specifically, we travel along R and always prefer to branch on an extremal path. When we close the cycle, we obtain a boundary cycle R_1. Then we visit R again, starting at an extremal point that is not on R_1. Again we prefer to branch on extremal paths. This will give a different cycle. The two cycles are inner and outer boundaries. For an example, see Figure 13 (ii).

With the correct inner and outer boundaries of F', we are now ready to recover the real boundaries of F. We simply put back the deleted bisectors. Indeed, the boundaries of F' are the boundaries of F plus all the bisectors and SPM-vertices. Thus we remove the bisectors from the boundaries of F' and obtain $k+1$ cycles that are the real boundaries.

Theorem 8 *Given a closed polygonal domain F in the plane, with k simple polygonal obstacles inside, the boundary detection algorithm will find the boundary of the region correctly.*

2.4 Simulations

We performed extensive simulations in various scenarios, with the goal to evaluate the performance of the algorithm with respect to the network topology, node density and distribution, etc. We particularly note that our method works well even in cases of very low average degree, such as less than 10, or even as low as 6 in some models. Degree 6 has been shown to be optimal for mobile networks [81].

2.4.1 Effect of node distribution and density

For each figure in this part, we show a pink circle in the upper left corner to illustrate the communication range of the sensor field.

2.4.1.1 Random distribution of sensors

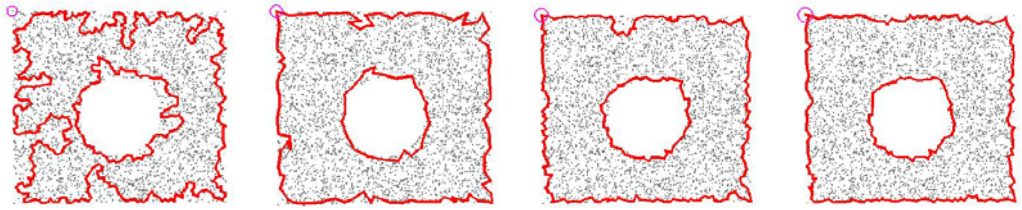
In this group of experiments, we randomly place, according to a uniform distribution, 3500 nodes in a square region with one hole. The average degree of the graph is varied by adjusting the communication radius. Figure 14 shows the results using our method. As expected, as the average degree of the network increases, the performance of the algorithm improves.

The unsatisfactory result in Figure 14(i) is due to insufficient connectivity. The average degree is 7; however, the random distribution tends to have clustered nodes and holes. Thus, in sparse regions some nodes are incorrectly judged to be extremal, and the final outer boundary is then required to pass through these mistaken extremal nodes. When the average degree reaches 10, the communication graph is better connected, and the results improve.

For comparison, SHAWN [68] did not find a starting solution (a "flower") for cases (i) and (ii); it found apparently correct boundaries for cases (iii) and (iv), indistinguishable from our results. The method of Funke and Klein [45] had difficulty with all 4 cases; see Figure 15(i), (ii).

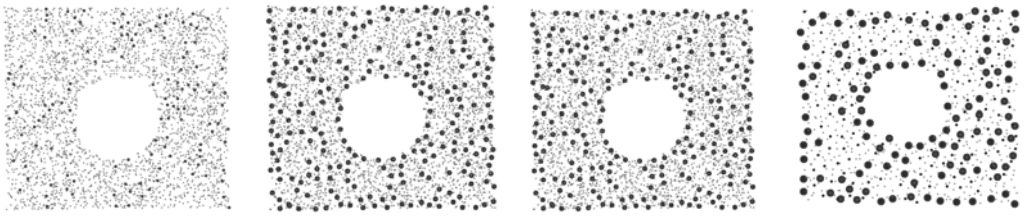
Connectivity is necessary for computing the shortest path tree and determining cuts, etc. In fact, this low-degree graph with insufficient connectivity is the major troubling issue for prior boundary detection methods. Since our method only requires the communication graph, we can apply some simple strategies to increase artificially the average degree. For a disconnected network, we use the largest connected component of the graph to build our shortest path tree. Then we artificially enlarge the communication radius by taking 2-hop/3-hop neighbors as "fake" 1-hop neighbors. In this way, the connectivity of the graph will be ameliorated and the results will be improved correspondingly. Figure 16 shows the improvement by this simple strategy. The result using 3-hop neighbors has fewer incorrectly marked extremal nodes, and the final boundary is in good shape except that the boundary

cycle is not very tight. This is understandable since we make the communication range artificially larger, so that more nodes are "on" the boundary now.



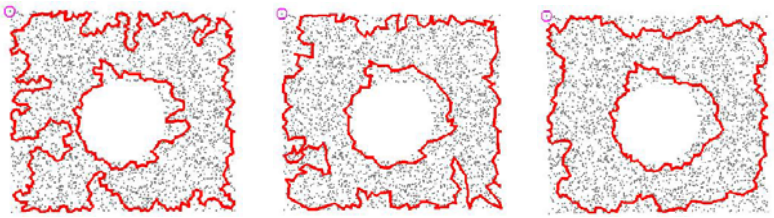
(i) (ii) (iii) (iv)

Figure 14: Uniformly distributed sensor field. (i) the average degree is 7; (ii) the average degree is 10; (iii) the average degree is 13; (iv) the average degree is 16.



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

Figure 15: Results of [45]: (i) uniform distribution, average degree 7; (ii) uniform, average degree 16; (iii) perturbed grid, average degree 8; (iv) sparse example (842 nodes), average degree 7.



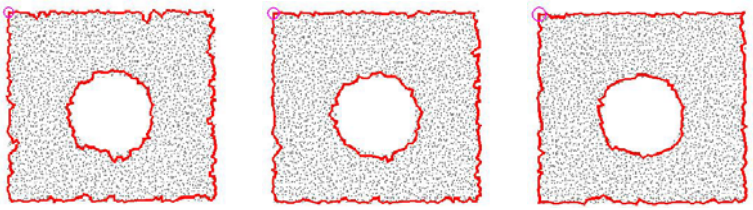
(i) (ii) (iii)

Figure 16: Using 2-hop/3-hop neighbors as fake 1-hop neighbors to improve the performance in the low average degree case. (i) based on the original neighbors; (ii) using 2-hop neighbors as fake 1-hop neighbors; (iii) using 3-hop neighbors as fake 1-hop neighbors.

2.4.1.2 Grid with random perturbation

In this simulation, we put about 3500 nodes on a grid and then perturbed each point by a random shift. In particular, for each original grid node we create two random numbers modulo the length and the width of each block of the grid, the use these two small numbers to perturb the positions of the nodes. This distribution may be a good approximation of manual deployments of sensors; it also gives an alternative means of modeling "uniform" distributions, while avoiding clusters and "holes" that can arise from the usual continuous uniform distribution or Poisson process. Refer to Figure 49. Our method gives very good results for graphs with average degree 6 or more.

SHAWN [68] found good boundaries for (ii)-(iii), but did not find a starting solution for (i). The method of Funke and Klein [45] did well for case (iii) but less well for the lower degree cases; see Figure 15(iii).



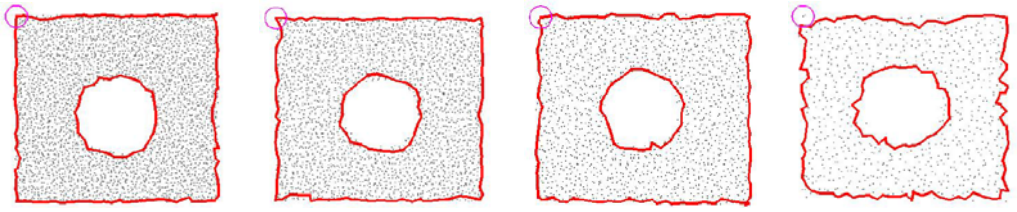
(i) (ii) (iii)

Figure 17: Results for randomly perturbed grids. (i) the average degree is 6; (ii) the average degree is 8; (iii) the average degree is 12.

2.4.1.3 Low density, sparse graphs

In this group of experiments (Figure 18), we scatter nodes in a square region with one hole. In order to guarantee good connectivity, the nodes are distributed on a randomly perturbed grid. The leftmost image of Figure 18 has about 3500 nodes; then, from left to right, each example has about 1000 fewer nodes, becoming more and more sparse. Our experiments show that if we fix the communication radius, and decrease the density of nodes, our method performs well, even for low density or sparse graphs (Figure 18(iv)), as long as the average degree is at about 7 or more.

SHAWN [68] did not find a starting solution for case (iv); it found good boundaries for cases (i)-(iii). The method of Funke and Klein [45] also performs well in cases (i)-(iii), but it performs less well in the lowest-degree case (iv); see Figure 15(iv).

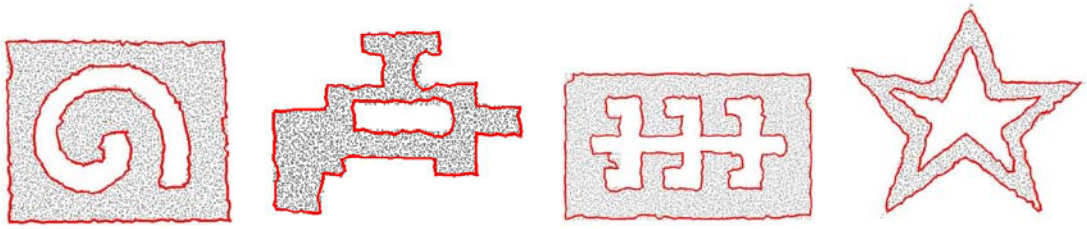


(i) (ii) (iii) (iv)

Figure 18: Results when the density of the graph decreases. (i) 3443 nodes and the average degree is 35; (ii) 2628 nodes and the average degree is 25; (iii) 1742 nodes and the average degree is 16; (iv) 842 nodes and the average degree is 7.

2.4.2 More examples

In Figure 19, we illustrate more results for a variety of more intricate geometries, including a spiral shape, a floorplan, etc.



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

Figure 19: Results for more interesting examples. (i) A spiral shape with 5040 nodes and the average degree is 21; (ii) A building floor shape with 3420 nodes and the average degree is 20; (iii) A cubicle shape in an office with 6833 nodes and the average degree is 17; (iv) A double star shape with 2350 nodes and the average degree is 17.

2.4.3 Further discussion

While our new boundary detection algorithm provably finds boundaries in the continuous case (Section 2.3), in discrete sensors networks several implementation issues arise.

First, even for a given homotopy type, there need not be a unique shortest path between two nodes. Thus, the boundary cycle discovered by our algorithm, as shown in the simulations, may not tightly surround the real boundaries. Currently, we have two approaches to improve it. One is to make use of the fact that the nodes with lower degree are more likely to be on the boundary; thus, we implemented a preferential scheme for low-degree nodes when computing shortest paths. Another approach is to use an iterative method to find more extremal nodes, and then refine the boundary; this can also help to address the issue that several extremal points may have the same positions because we use hop counts to approximate true distances.

Second, deciding the correct orderings of the extremal nodes requires some care. In the continuous case, extremal nodes project to their nearest node in the rough inner boundary, resulting in a consistent ordering of extremal nodes, as shown in Section 2.3. In the discrete case, since we use hop count to approximate the true distance, it is possible that different extremal points are mapped to the same position on the inner boundary, obscuring their ordering. Again, by using an iterative procedure, we delete all the extremal nodes with duplicate positions except one and then iteratively find more extremal points and refine the boundary gradually.

In practice, the sensor nodes often know some partial location information or relative angular information. Such positional information can help to improve the performance of our boundary detection algorithm. For example, if the nodes have knowledge of a universal north direction, it is easier to distinguish the extremal nodes in the interior and exterior of rough boundary. Also, if we have estimated distance or other rough localization information, other than pure hop count, the procedure to find shortest paths will become more reliable.

Finally, our method discussed until now assumes a sensor field with holes. We remark that the case with no holes can be solved as well. If a network has no hole, we discover this fact, since the network has no cut. In addition, the rough inner boundary degenerates to the root node. Then, we can discover the extremal nodes

and connect them to find the outer boundary.

2.5 Conclusion

We propose a simple and distributed topological approach to recognize the boundaries of a sensor field using only their communication graph. We not only discover boundary nodes but we also connect them in a meaningful way. In addition, we can distinguish between inner boundaries and the outer boundary. We prove that the algorithm works correctly in a continuous domain and demonstrate the effectiveness via extensive simulation for various networks with different density and topology.

Chapter 3

Geodesic Delaunay Triangulations and Witness Complexes in the Plane

3.1 Introduction

In the chapter we focus on the special case of bounded domains in the plane – a setting which already raises numerous questions and finds important applications in sensor networks. We make the novel claim that resorting to an intrinsic metric instead of the Euclidean metric can result in very significant reductions in terms of the number of samples required to recover the homotopy type of a bounded domain – an especially appealing fact in the context of resource-constrained nodes used in sensor networks. To this end, we introduce a new quantity, called the homotopy feature size, or hfs for short, which measures the size of the smallest topological feature (hole in this case) of the considered planar domain X. Specifically, given a point $x \in X$, $\mathrm{hfs}(x)$ is defined as half the length of the shortest loop through x that is not null-homotopic in X – see Figure 3 for an illustration. In particular, $\mathrm{hfs}(x)$ is infinite whenever x lies in a simply connected component of X. In contrast with previous quantities, hfs depends essentially on the global topology of X, and it is only marginally influenced by the local geometry of the domain boundary. Under the assumption that X has Lipschitz boundaries (the actual Lipschitz constant being unimportant in our context), we show that hfs is well-defined, positive, and 1-Lipschitz in the intrinsic metric. Moreover, if L is a geodesic εhfs-sample of X, for

some $\varepsilon < \frac{1}{3}$, then the cover of X formed by the geodesic Voronoi cells of the points of L satisfies the conditions of the Nerve theorem [16, 92], and therefore its dual Delaunay complex $D_X(L)$ is homotopy equivalent to X. By geodesic εhfs-sample of X, we mean that every point $x \in X$ is at a finite geodesic distance to L, bounded from above by $\varepsilon\,\mathrm{hfs}(x)$. In the particular case when X is simply connected, our sampling condition only requires that L have at least one point on each connected component of X, regardless of the local geometry of X. In the general case, our sampling condition can be satisfied by placing a constant number of landmarks around each hole of X, and a number of landmarks in the remaining parts of X that is logarithmic in the ratio of the geodesic diameter of X to the geodesic perimeter of its holes. This is rather independent of the local geometry of the boundary ∂X and can result in selecting far fewer landmarks than required by any of the earlier sampling conditions that guarantee topology capture.

The homotopy feature size is closely related to the concept of normal injectivity radius in Riemannian geometry. We stress this relationship in the chapter, by showing that, for all point $x \in X$, $\mathrm{hfs}(x)$ is equal to the geodesic distance from x to its cut-locus in X. This result also suggests a simple procedure for estimating $\mathrm{hfs}(x)$ at any point $x \in X$. Using this procedure, we devise a greedy algorithm for generating εhfs-samples of any given Lipschitz planar domain X, based on a packing strategy. The size of the output lies within a constant factor of the optimal, the constant depending on the doubling dimension of X. Our algorithm relies on two oracles whose actual implementations depend on the application considered. We provide some implementations in the context of sensor networks, based on pre-existing distributed schemes [37, 89].

Finally, we focus on the structural properties of the so-called *geodesic witness complex*, an analog of the usual witness complex in the intrinsic metric. In many applications, computing $D_X(L)$ can be hard, due to the difficulty of checking whether three or more geodesic Voronoi cells have a common intersection. This is especially true in sensor networks, where the intersections between the Voronoi cells of the landmarks can only be sought for among the set of nodes W, due to the lack of further information on the underlying domain X. Therefore, it is convenient to replace $D_X(L)$ by the geodesic witness complex $\mathbb{C}_X^W(L)$, whose computation only requires

us to perform geodesic distance comparisons, instead of locating points equidistant to multiple landmarks. Assuming that the geodesic distance can be computed exactly, we prove an analog of de Silva's theorem [29], which states that $\mathbb{C}_X^W(L)$ is included in $D_X(L)$ under some mild sampling conditions. We also prove an analog of Lemma 3.1 of [57], which states that a relaxed version of $\mathbb{C}_X^W(L)$ by allowing a simplex to be ν-witnessed by w if its vertices belong to the $\nu+1$ nearest landmarks of w, denoted by $\mathbb{C}_{X,\nu}^W(L)$, contains $D_X(L)$ under similar conditions. Unfortunately, as pointed out in [57], it is often the case that neither of them coincides with $D_X(L)$. However, taking advantage of the fact that $D_X(L)$ is sandwiched between $\mathbb{C}_X^W(L)$ and $\mathbb{C}_{X,\nu}^W(L)$, we show that computing the persistent homology between $\mathbb{C}_X^W(L)$ and $\mathbb{C}_{X,\nu}^W(L)$ gives the homology of $D_X(L)$. This allows to retrieve the homology of X without computing $D_X(L)$ in practice. Similar results have been proved for other types of filtrations [25, 27] and used in the context of sensor networks [52]. However, to the best of our knowledge, our result is the first one of this type for the witness complex filtration.

3.2 The intrinsic metric

Let $I=[0,1]$. The ambient space is R^2, endowed with the Euclidean metric, noted d_E. Given a subset X of R^2, $\mathring{X}$, $\overline{X}$, and ∂X, stand respectively for the interior, closure, and boundary of X. Given $x\in R^2$ and $r\in R_+$, $B_E(x,r)$ denotes the Euclidean open ball of radius r about x. Finally, S^1, $R\times\{0\}$, and R_+^2, denote respectively the unit circle, the abcissa line, and the closed upper half-plane in R^2.

Paths and loops. Given a topological space X, a *path* in X is a continuous map $I\to X$. For all $a,b\in I$ ($a\leq b$), we call $\gamma_{|[a,b]}$ the path $s\mapsto\gamma(a+s(b-a))$, which can be viewed as the restriction of γ to the segment $[a,b]$. In addition, $\overline{\gamma}$ denotes the inverse path $s\mapsto\gamma(1-s)$. Given another path $\gamma':I\to X$ such that $\gamma'(0)=\gamma(1)$, we call $\gamma\cdot\gamma'$ their concatenation, defined by $\gamma\cdot\gamma'(s)=\gamma(2s)$ for $0\leq s\leq\frac{1}{2}$ and $\gamma\cdot\gamma'(s)=\gamma'(2s-1)$ for $\frac{1}{2}\leq s\leq 1$. Given a point $x\in X$, a *loop* through x in X is a path γ in X that starts and ends at x, *i.e.* such that $\gamma(0)=\gamma(1)=x$. For simplicity, we write $\gamma:(I,\partial I)\to(X,x)$. Note that γ can also be seen as a continuous map from the unit circle to X, and we write $\gamma:(S^1,1)\to(X,x)$ to specify that $\gamma(1)=x$.

To any loop $\gamma : S^1 \to S^1$ corresponds a unique integer $\deg\gamma \in Z$, called the *degree* of γ, such that $\deg\gamma = 0$ if γ is a constant map $S^1 \to \{x\}$, and $\deg(\gamma\cdot\gamma') = \deg\gamma + \deg\gamma'$ for any loop $\gamma' : S^1 \to S^1$ satisfying $\gamma'(0) = \gamma(1)$. Moreover, it can be proved that $\deg\gamma = \deg\gamma'$ iff γ and γ' are homotopic in S^1 [58, Thm. 1.7], so that $\deg\gamma$ is a unique descriptor of the homotopy class of the loop γ. A similar concept exists for loops in the plane. Given $\gamma : S^1 \to R^2$ and $x \in R^2 \setminus \gamma(S^1)$, consider the map $\gamma_x = \pi_x \circ \gamma : S^1 \to S^1$, where $\pi_x : R^2 \setminus \{x\} \to S^1$ is the radial projection: $\pi_x(y) = \frac{y-x}{\|y-x\|}$. Since π_x is continuous over $R^2 \setminus \{x\}$, γ_x is a continuous loop in S^1. We then define the degree (aka winding number) of γ with respect to x as: $\deg_x\gamma = \deg\gamma_x$. If $\Gamma : S^1 \times I \to R^2 \setminus \{x\}$ is a homotopy between two loops γ, γ' in $R^2 \setminus \{x\}$, then $\pi_x \circ \Gamma$ is a homotopy between $\pi_x \circ \gamma$ and $\pi_x \circ \gamma'$ in S^1, and therefore we have: $\deg_x\gamma = \deg(\pi_x \circ \gamma) = \deg(\pi_x \circ \gamma') = \deg_x\gamma'$. As a result,

Corollary 9 *For any point $x \in R^2$ and any loops $\gamma, \gamma' : S^1 \to R^2 \setminus \{x\}$ that are homotopic in $R^2 \setminus \{x\}$, we have $\deg_x\gamma = \deg_x\gamma'$. In particular, if γ or γ' is constant, then $\deg_x\gamma = \deg_x\gamma' = 0$.*

Length structures and Lipschitz planar domains. A good introduction to length spaces can be found in [20, Chap. 2]. Every subset X of R^2 inherits a length structure from R^2, where admissible paths are all continuous paths $I \to X$, and where the length of a path γ is defined by: $|\gamma| = \sup\{\sum_{i=0}^{n-1} \mathrm{d}_E(\gamma(t_i), \gamma(t_{i+1})),\ n \in N\ ,\ 0 = t_0 \le t_1 \le \cdots \le t_n = 1\}$, the supremum being taken over all decompositions of I into an arbitrary (finite) number of intervals. We clearly have $|\overline{\gamma}| = |\gamma|$. However, $|\gamma|$ is not always finite. Take for instance Koch's snowflake, a fractal curve defined as the limit of a sequence of polygonal curves in the plane. It can be easily shown that, at each iteration of the construction, the length of the curve is multiplied by $\frac{4}{3}$, so that the length of the limit curve is infinite. We say that $\gamma : I \to X$ is a *rectifiable* path if its length $|\gamma|$ is finite.

We make X into a length space by defining an *intrinsic* (or *geodesic*) metric d_X as follows: $\forall x, y \in X$, $\mathrm{d}_X(x,y) = \inf\{|\gamma|,\ \gamma : I \to X,\ \gamma(0) = x,\ \gamma(1) = y\}$, the infimum being taken over all paths from x to y in X. Clearly, $\mathrm{d}_X(x,y) = +\infty$ whenever x, y belong to different path-connected components of X. However, the converse is not always true. Take for instance a domain X made of two disjoint disks connected by

Koch's snowflake: if x, y belong to different disks, then all curves connecting x and y go through Koch's snowflake and therefore have infinite length. As a consequence, the intrinsic topology induced by d_X on X can be different from the Euclidean topology induced by d_E. This is a critical issue because the geodesic Voronoi diagram is bound to the intrinsic metric, whereas our goal is to retrieve the homotopy type of X in the extrinsic metric. Another issue is that not all pairs of points $x, y \in X$ with $\mathrm{d}_X(x,y) < +\infty$ may have a shortest path in X, *i.e.* a path $\gamma : I \to X$ such that $\gamma(0) = x$, $\gamma(1) = y$, and $|\gamma| = \mathrm{d}_X(x,y)$. Take for instance two diametral points on the boundary of the unit closed disk, to which the closed disk of radius $\frac{1}{2}$ has been removed. These issues lead us to consider the special case of Lipschitz domains:

Definition 10 *A* Lipschitz domain *in the plane is a compact embedded topological 2-submanifold of $\mathbb{R}^2$ with Lipschitz boundary. Formally, it is a compact subset X of $\mathbb{R}^2$ such that, for all point $x \in \partial X$, there exists a neighborhood V_x in $\mathbb{R}^2$ and a Lipschitz homeomorphism $\phi_x : \mathbb{R}^2 \to \mathbb{R}^2$, such that $\phi_x(0) = x$, $\phi_x(\mathbb{R} \times \{0\}) \cap V_x = \partial X \cap V_x$, and $\phi_x(\mathbb{R}^2_+) \cap V_x = X \cap V_x$.*

Observe that, for any neighborhood $V'_x \subseteq V_x$, we also have $\phi_x(0) = x$, $\phi_x(\mathbb{R} \times \{0\}) \cap V'_x = \partial X \cap V'_x$, and $\phi_x(\mathbb{R}^2_+) \cap V'_x = X \cap V'_x$. Therefore, V_x can be assumed to be arbitrarily small. Moreover, since $\phi_x(0) = x$ and ϕ_x is continuous, $\phi_x^{-1}(V_x)$ is a neighborhood of the origin in $\mathbb{R}^2$, hence it contains an open Euclidean disk B about the origin. By taking $\phi(B)$ as the new neighborhood V_x, we ensure that $\phi_x^{-1}(X \cap V_x)$ is the intersection of $\mathbb{R}^2_+$ with the open disk B, and therefore that it is convex.

Note that the actual Lipschitz constants of the charts ϕ_x in Definition 10 are unimportant: only the fact that the ϕ_x are Lipschitz counts. This makes the class of Lipschitz planar domains quite large: in particular, it contains all planar domains with piecewise-analytic boundaries. Moreover, the pathologies described above cannot occur on a Lipschitz domain, by the following theorem:

Theorem 11 *If X is a Lipschitz domain in the plane, then:*

(i) the intrinsic topology coincides with the Euclidean topology on X;

(ii) every sequence of paths with uniformly bounded length contains a uniformly converging subsequence; therefore, all points $x, y \in X$ such that $\mathrm{d}_X(x,y) < +\infty$ have a shortest path in X;

(iii) for any path $\gamma : I \to X$ *and any real number* $\varepsilon > 0$*, there exists a rectifiable path* $\gamma_\varepsilon : I \to X$*, homotopic to* γ *relative*[1] *to* ∂I *in* X*, such that* $\max_{s\in I} \min_{t\in I} \mathrm{d}_X(\gamma_\varepsilon(s), \gamma(t)) < \varepsilon$.

The proof of the theorem is given in the full version of this work [51]. It relies on the following facts: given a point $x \in \mathring{X}$, there is a small convex neighborhood $V_x \subseteq X$ inside which any given arc can be continuously deformed into a rectifiable arc. Now, given a point $x \in \partial X$, there is no such neighborhood as above. However, Definition 10 provides us with a neighborhood V_x and a Lipschitz homeomorphism ϕ_x such that $\phi_x^{-1}(V_x \cap X)$ is convex. Then, inside $\phi_x^{-1}(V_x \cap X)$, we can deform any given arc into a rectifiable arc, whose image through ϕ_x is rectifiable and included in X.

3.3 The homotopy feature size

Definition 12 *Given a Lipschitz planar domain* X *and a point* $x \in X$*, the* homotopy feature size *of* X *at* x *is the quantity:* $\mathrm{hfs}(x) = \frac{1}{2} \inf\{|\gamma|,\ \gamma : (S^1, 1) \to (X, x)$ *non null-homotopic in* $X\}$.

As illustrated in Figure 3, the resort to the intrinsic metric makes the homotopy feature size rather insensitive to the local geometry of the domain X. We will prove in Section 3.5.1 that $\mathrm{hfs}(x)$ equals the geodesic distance from x to its cut-locus in X, thus showing the relationship that exists between the homotopy feature size and the concept of normal injectivity radius in Riemannian geometry.

Lemma 13 *Let* X *be a Lipschitz domain in the plane.*

(i) Given a point $x \in X$*, if the path-connected component of* X *that contains* x *is simply connected, then* $\mathrm{hfs}(x) = +\infty$*. Else,* $\mathrm{hfs}(x) < +\infty$*, and there exists a non null-homotopic rectifiable loop* $\gamma : (S^1, 1) \to (X, x)$ *such that* $\mathrm{hfs}(x) = \frac{1}{2}|\gamma| > 0$.

(ii) The map $x \mapsto \mathrm{hfs}(x)$ *is* 1*-Lipschitz in the intrinsic metric. As a consequence, it is continuous for the Euclidean topology, by Theorem 11, and* $\mathrm{hfs}(X) = \inf\{\mathrm{hfs}(x),\ x \in X\}$ *is positive.*

[1] *This means that the homotopy between* γ_ε *and* γ *is constant over* $\partial I = \{0, 1\}$.

The proof of assertion (i) considers an arbitrary sequence of non null-homotopic rectifiable loops through x, whose lengths converge towards $2\,\mathrm{hfs}(x)$, and it applies Theorem 11 (ii) to this sequence. The proof of assertion (ii) takes two points x, y lying in a same path-connected component of X that is not simply connected, and it considers the non null-homotopic loop γ_x through x provided by assertion (i), as well as a shortest path γ from y to x. Then, $\gamma_y = \gamma \cdot \gamma_x \cdot \overline{\gamma}$ is a non null-homotopic rectifiable loop through y, of length $|\gamma_y| = 2\,\mathrm{hfs}(x) + 2\,\mathrm{d}_X(x,y)$. It follows that $\mathrm{hfs}(y) \leq \frac{1}{2}|\gamma_y| = \mathrm{hfs}(x) + \mathrm{d}_X(x,y)$.

Lemma 14 *Let X be a Lipschitz domain in the plane. For all point $x \in X$, every loop inside the geodesic open ball $B_X(x, \mathrm{hfs}(x))$ is null-homotopic in X.*

Intuitively, a geodesic ball of center x and radius less than $\mathrm{hfs}(x)$ cannot enclose any hole of X, therefore every loop inside such a ball must be null-homotopic in X. A formal proof is given in the full version of the work [51]. Note that Lemma 14 does not imply that $B_X(x, \mathrm{hfs}(x))$ itself is contractible. This fact is true nevertheless, but its proof requires some more work.

3.4 Structural results

Given a Lipschitz domain X in the plane, and a set of landmarks $L \subset X$ that is dense enough with respect to the homotopy feature size of X, we show in Section 3.4.1 that the geodesic Delaunay triangulation $D_X(L)$ has the same homotopy type as X (Theorem 15). Furthermore, for any set of witnesses $W \subseteq X$ that is dense enough compared to L, we prove in Section 3.4.2 that $D_X(L)$ is sandwiched between the geodesic witness complex $\mathbb{C}_X^W(L)$ and its relaxed version $\mathbb{C}_{X,\nu}^W(L)$ (Theorems 20 and 21).

3.4.1 Geodesic Delaunay triangulation

Consider a domain $X \subseteq R^2$ and a finite set of sites $L \subset X$. The *geodesic Voronoi cell* of a site p is the locus of the points $x \in X$ satisfying $\mathrm{d}_X(x,p) \leq \mathrm{d}_X(x,q)$ for all $q \in L$. The *geodesic Voronoi diagram* of L in X, or $V_X(L)$ for short, is

the cellular decomposition of X formed by the geodesic Voronoi cells of the sites. The nerve of $V_X(L)$ is called the *geodesic Delaunay triangulation* of L in X, noted $D_X(L)$. The face of $V_X(L)$ dual to a given simplex $\sigma \in D_X(L)$ is noted $\mathrm{V}_X(\sigma)$.

Consider now a Lipschitz planar domain X, and a finite set of sites $L \subset X$ that is a *geodesic* εhfs-*sample* of X, for some $\varepsilon < \frac{1}{3}$. This means that, for all point $x \in X$, the geodesic distance from x to L is finite and at most $\varepsilon\,\mathrm{hfs}(x)$. Note that L has at least one point in every path-connected component of X, because geodesic distances to L are required to be finite. We will see how to generate such point sets in Section 3.5.2.

Theorem 15 *If X is a Lipschitz domain in the plane, and if L is a geodesic* εhfs-*sample of X, for some $\varepsilon < \frac{1}{3}$, then $D_X(L)$ and X are homotopy equivalent.*

The rest of Section 3.4.1 is devoted to the proof of Theorem 15, which uses the Nerve theorem:

Theorem 16 (from [16, 92], see also Thm. 10.7 of [12]) *Let U be a finite closed cover of X, such that the intersection of any collection of elements of U is either empty or contractible. Then, the nerve of U is homotopy equivalent to X.*

In our case, we set U to be the collection of the geodesic Voronoi cells: $U = \{V_X(p),\ p \in L\}$. The nerve of this collection is precisely the geodesic Delaunay triangulation $D_X(L)$. Thus, Theorem 16 reduces the proof of Theorem 15 to showing that the intersection of any arbitrary number of cells of $V_X(L)$ is empty or contractible. We first show that the geodesic Voronoi cells are contractible:

Lemma 17 *Under the hypotheses of Theorem 15, every cell of $V_X(L)$ is contractible.*

Proof. Let $p \in L$. We first show that $V_X(p)$ is path-connected. Let $x \in V_X(p)$, and let $\gamma : I \to X$ be a shortest path from p to x in X. Such a path γ exists by Theorem 11 (ii), since x and p lie in the same path-connected component of X, $\mathrm{d}_X(x,p)$ being finite due to the fact that L is a geodesic εhfs-sample of X. We will show that $\gamma(I) \subseteq V_X(p)$. Assume for a contradiction that $\gamma(s) \notin V_X(p)$ for some $s \in I$. This means that there exists a point $q \in L \setminus \{p\}$ such that $\mathrm{d}_X(\gamma(s), q) <$

$\mathrm{d}_X(\gamma(s),p)$. By the triangle inequality, we have $\mathrm{d}_X(q,x) \leq \mathrm{d}_X(q,\gamma(s)) + \mathrm{d}_X(\gamma(s),x)$, where $\mathrm{d}_X(q,\gamma(s)) < \mathrm{d}_X(p,\gamma(s)) \leq |\gamma_{|[0,s]}|$ and $\mathrm{d}_X(\gamma(s),x) \leq |\gamma_{|[s,1]}|$. Hence, we have $\mathrm{d}_X(q,x) < |\gamma_{|[0,s]}| + |\gamma_{|[s,1]}| = |\gamma| = \mathrm{d}_X(p,x)$, which contradicts the assumption that $x \in V_X(p)$. Therefore, $\gamma(I) \subseteq V_X(p)$, and x is path-connected to p in $V_X(p)$. This shows that $V_X(p)$ is path-connected.

Assume now for a contradiction that $V_X(p)$ is not simply connected. Then, since $V_X(p) \subseteq X$ is a bounded subset of R^2, its complement in R^2 has at least two path-connected components, only one of which is unbounded, by the Alexander duality – see *e.g.* [58, Thm. 3.44]. Let H be a bounded path-connected component of $R^2 \setminus V_X(p)$. H can be viewed as a hole in $V_X(p)$. We claim that H is included in X. Indeed, consider a loop $\gamma : S^1 \to V_X(p)$ that winds around H – such a loop exists since H is bounded by $V_X(p)$. Take any point $x \in V_X(p)$. For all $y \in V_X(p)$, we have $\mathrm{d}_X(x,y) \leq \mathrm{d}_X(x,p) + \mathrm{d}_X(p,y) \leq \varepsilon\,\mathrm{hfs}(x) + \varepsilon\,\mathrm{hfs}(y)$, which is at most $\frac{2\varepsilon}{1-\varepsilon}\mathrm{hfs}(x)$ since hfs is 1-Lipschitz in the intrinsic metric (Lemma 13 (ii)). Thus, $V_X(p)$ is included in the geodesic closed ball $B_X(x, \frac{2\varepsilon}{1-\varepsilon}\,\mathrm{hfs}(x))$, where $\frac{2\varepsilon}{1-\varepsilon} < 1$ since $\varepsilon < \frac{1}{3}$. Therefore, $\gamma : S^1 \to V_X(p)$ is null-homotopic in X, by Lemma 14. Let $\Gamma : S^1 \times I \to X$ be a homotopy between γ and a constant map in X. For any point $z \in H$, we have $\deg_z \gamma \neq 0$ since the loop γ winds around H. If z did not belong to $\Gamma(S^1 \times I)$, then Γ would be a homotopy between γ and a constant map in $R^2 \setminus \{z\}$, thus by Corollary 9 we would have $\deg_z \gamma = 0$, thereby raising a contradiction. Hence, $\Gamma(S^1 \times I)$ contains all the points of hole H, which is therefore included in X.

As a consequence, the hole is caused by the presence of some sites of $L \setminus \{p\}$, whose geodesic Voronoi cells form H. Assume without loss of generality that there is only one such site q. We then have $V_X(q) = \overline{H}$, and $\partial H = V_X(q) \cap V_X(p)$. Consider the Euclidean ray $[p,q)$, and call x its first point of intersection with ∂H beyond q. The line segment $[q,x]$ is included in $\overline{H} \subseteq X$, therefore we have $\mathrm{d}_X(x,q) = \mathrm{d}_E(x,q)$, which yields: $\mathrm{d}_X(x,p) \geq \mathrm{d}_E(x,p) = \mathrm{d}_E(x,q) + \mathrm{d}_E(q,p) = \mathrm{d}_X(x,q) + \mathrm{d}_E(q,p) > \mathrm{d}_X(x,q)$. This contradicts the fact that x belongs to ∂H and hence to $V_X(p)$. Thus, $V_X(p)$ is simply connected. Since it is also path-connected, it is contractible. □

By very similar arguments, we can prove that the union of any two intersecting

cells of $V_X(L)$ is contractible. It follows then from Lemma 17 and from the following classical result of algebraic topology that their intersection is also contractible:

Lemma 18

(i) The intersection of any k simply connected subsets of R^2 is either empty or simply connected.

(ii) If X, Y are path-connected subsets of R^2 such that $X \cup Y$ is simply connected, then $X \cap Y$ is either empty or path-connected.

We will now extend the above results to the intersections of arbitrary numbers of cells of $V_X(p)$, thereby concluding the proof of Theorem 15:

Lemma 19 *Under the hypotheses of Theorem 15, for any k sites $p_1, \cdots, p_k \in L$, the intersection $V_X(p_1) \cap \cdots \cap V_X(p_k)$ is either empty or contractible.*

Proof. The proof is by induction on k. Cases $k = 1$ and $k = 2$ have just been proved. Assume now that the result is true up to some $k \geq 2$, and consider $k+1$ sites $p_1, \cdots, p_{k+1} \in L$ such that $V_X(p_1) \cap \cdots \cap V_X(p_{k+1}) \neq \emptyset$. Notice first that $V_X(p_1) \cap \cdots \cap V_X(p_{k+1})$ is the intersection of $\bigcap_{i=1}^{k} V_X(p_i)$ with $V_X(p_{k+1})$, which by the induction hypothesis are simply connected. Hence, their intersection $V_X(p_1) \cap \cdots \cap V_X(p_{k+1})$ is also simply connected, by Lemma 18 (i). Observe now that $\left(\bigcap_{i=1}^{k} V_X(p_i)\right) \cup V_X(p_{k+1})$ can be rewritten as $\bigcap_{i=1}^{k} (V_X(p_i) \cup V_X(p_{k+1}))$. By the induction hypothesis (more precisely, according to the case $k = 2$), every $V_X(p_i) \cup V_X(p_{k+1})$ is simply connected, hence so is $\bigcap_{i=1}^{k} (V_X(p_i) \cup V_X(p_{k+1}))$, by Lemma 18 (i). It follows then from Lemma 18 (ii) that the intersection $V_X(p_1) \cap \cdots \cap V_X(p_{k+1})$ is path-connected, since by induction both $\bigcap_{i=1}^{k} V_X(p_i)$ and $V_X(p_{k+1})$ are, and since their union is simply connected. □

3.4.2 Geodesic witness complex

Consider a domain $X \subseteq R^2$, as well as two finite subsets L and W. Given a point $w \in W$ and a simplex $\sigma = [p_0, \cdots, p_l]$ with vertices in L, w is a *witness* of σ if for all $i = 0, \cdots, l$, $\mathrm{d}_X(w, p_i)$ is finite and bounded from above by $\mathrm{d}_X(w, q)$ for every $q \in L \setminus \{p_0, \cdots, p_l\}$. Observe that w may only witness simplices whose vertices lie

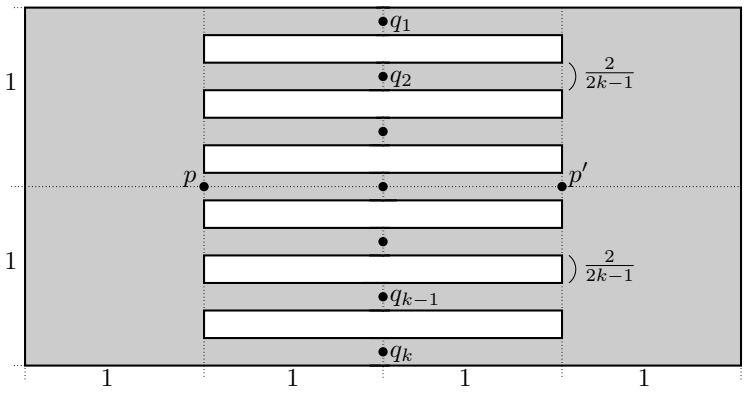


Figure 20: The size of $\{q_1, \cdots, q_k\}$ is $\frac{k}{2}$ times that of $\{p, p'\}$, although both are sparse hfs-samples.

in the same path-connected component of X. The *geodesic witness complex* of L relative to W, or $\mathbb{C}_X^W(L)$ for short, is the maximal abstract simplicial complex with vertices in L, whose faces are witnessed by points of W. The fact that $\mathbb{C}_X^W(L)$ is an abstract simplicial complex means that a simplex belongs to the complex only if all its faces do. In the sequel, W will be referred to as the set of witnesses, and L as the set of landmarks.

Our first result is an analog of de Silva's theorem [29] in the intrinsic metric. The proof uses the same machinery as in [7], and it relies on the intuitive fact that, when the set L is a geodesic εhfs-sample of X, the geodesic distances between a point $x \in X$ and its k nearest landmarks in the intrinsic metric are at most $4^k \varepsilon \, \mathrm{hfs}(x)$, the exponent coming from the fact that hfs is 1-Lispchitz.

Theorem 20 *Let X be a Lipschitz domain in the plane, and L a geodesic* εhfs*-sample of X. If $\varepsilon \leq \frac{1}{4^{k+1}}$, for some integer $k \geq 0$, then the k-skeleton of $\mathbb{C}_X^W(L)$ is included in $D_X(L)$ for all $W \subseteq X$.*

Our next result is an analog of Theorem 3.2 of [57], whose proof relies on a simple packing argument. It involves a relaxed version of the witness complex, defined as follows. Given an integer $\nu \geq 0$, a simplex σ is ν-witnessed by $w \in W$ if the vertices of σ belong to the $\nu + 1$ landmarks closest to w in the intrinsic metric. The geodesic ν-witness complex of L relative to W, or $\mathbb{C}_{X,\nu}^W(L)$ for short, is the maximum abstract simplicial complex made of ν-witnessed simplices. Its dimension is at most ν.

The theorem assumes that L is a $\frac{\varepsilon}{1+\varepsilon}$hfs-sparse geodesic εhfs-sample of X, which means that every pair of landmarks p,q satisfies: $\mathrm{d}_X(p,q) \geq \frac{\varepsilon}{1+\varepsilon}\min\{\mathrm{hfs}(p),\mathrm{hfs}(q)\}$. The bound on ε depends on the *doubling dimension* of (X,d_X), defined as the smallest integer d such that every geodesic closed ball can be covered by a union of 2^d geodesic closed balls of half its radius. The doubling dimension measures the shape complexity of X, and it can be arbitrarily large. An example is given in Figure 20, where the k geodesic balls $B_X(q_i,1)$ are included in their respective branches, and therefore are disjoint. Moreover, they are packed inside the ball $B_X(p,3)$, which therefore requires at least k geodesic unit balls to be covered, by a result of [67]. It follows that the doubling dimension of X is at least $\frac{1}{2}\log_2 k$, which can be made arbitrarily large.

Theorem 21 *Let X be a Lipschitz domain in the plane, of doubling dimension d. Let W be a geodesic δ*hfs*-sample of X, and L a geodesic ε*hfs*-sample of X that is also $\frac{\varepsilon}{1+\varepsilon}$*hfs*-sparse. If $\varepsilon+2\delta<1$, then, for any integer $\nu \geq 2^{d+1}\frac{(1+\delta/\varepsilon)(1+\varepsilon)}{1-\varepsilon-2\delta}$, $D_X(L)$ is included in $\mathbb{C}^W_{X,\nu}(L)$.*

Combining Theorems 20 and 21, we get that, whenever L and W are dense enough, $D_X(L)$ is sandwiched between $\mathbb{C}^W_X(L)$ and $\mathbb{C}^W_{X,\nu}(L)$, provided that ν is chosen sufficienly large. Our simulation results – see Section 3.6 – suggest that even small values of ν are sufficient in practice.

3.5 Algorithms

In this section, we describe high-level procedures for computing hfs, for generating geodesic εhfs-samples, and for computing the homology of a Lipschitz planar domain. Our algorithms rely essentially on two oracles, whose implementations depend on the application considered. Section 3.6 will be devoted to the implementation of such oracles on a sensor network.

3.5.1 Computing the homotopy feature size

The homotopy feature size is closely related to the concept of cut-locus. Given a path $\gamma : I \to X$, we call trajectory of γ the set $\gamma(I)$. If γ is a shortest path between

$x = \gamma(0)$ and $y = \gamma(1)$, then $\gamma(I)$ is called a shortest trajectory between x and y. Given a point $x \in X$, the cut-locus of x in X, or $\mathrm{CL}_X(x)$ for short, is the locus of the points of X having at least two distinct shortest trajectories to x in X. The geodesic distance from x to its cut-locus is denoted by $\mathrm{d}_X(x, \mathrm{CL}_X(x))$.

Lemma 22 *If X is a Lipschitz domain in the plane, then $\forall x \in X$, $\mathrm{hfs}(x) = \mathrm{d}_X(x, \mathrm{CL}_X(x))$.*

The proof is given in the full version of this work [51]. Lemma 22 suggests a simple way of computing hfs: given a point $x \in X$, grow a geodesic closed ball B about x, starting with a radius of zero and ending when B covers the path-connected component X_x of X containing x. Meanwhile, focus on the wavefront ∂B as the radius of B increases – this wavefront evolves as the iso-level sets of the map $y \mapsto \mathrm{d}_X(x,y)$. If at some stage the wavefront *self-intersects*, *i.e.* if there is a point $y \in \partial B$ with two or more distinct shortest trajectories to x, then interrupt the growing process and return the current value of the radius of B. Else, stop once B covers X_x and return $+\infty$.

By detecting the first self-intersection event in the growing process, the procedure finds a point of $\mathrm{CL}_X(x)$ closest to x in the intrinsic metric, and therefore it returns $\mathrm{d}_X(x, \mathrm{CL}_X(x))$, which by Lemma 22 is equal to $\mathrm{hfs}(x)$. The procedure relies on two oracles: the first one detects whether B covers X_x entirely; the second one detects whether the wavefront self-intersects at a given value r of the radius of B, or rather, between two given values $r_1 < r_2$ of the radius of B.

3.5.2 Generating geodesic εhfs-samples

Given a Lipschitz planar domain X and a parameter $\varepsilon > 0$, we use a greedy packing strategy to generate geodesic εhfs-samples of X. Initially, our algorithm selects an arbitrary point $p \in X$ and sets $L = \{p\}$. It also assigns to p the geodesic open ball B_p of center p and radius $\frac{\varepsilon}{1+\varepsilon}\,\mathrm{hfs}(p)$, where $\mathrm{hfs}(p)$ is estimated using the procedure of Section 3.5.1. If $\mathrm{hfs}(p) = +\infty$, then B_p coincides with the path-connected component of X containing p. Then, at each iteration, the algorithm selects an arbitrary point $q \in X \setminus \bigcup_{p \in L} \overline{B}_p$, and it inserts this point in L. It also

assigns a geodesic open ball B_q to q, as detailed above for p. The process stops when $X \setminus \bigcup_{p \in L} \overline{B}_p = \emptyset$.

The algorithm uses a variant of an oracle of Section 3.5.1, which can tell whether a given union of geodesic balls covers X, and return a point outside the union in the negative. Upon termination, every point $x \in X$ lies in some closed ball $\overline{B}_p$, and we have $\mathrm{d}_X(x,L) \leq \mathrm{d}_X(x,p) \leq \frac{\varepsilon}{1+\varepsilon}\,\mathrm{hfs}(p)$, which is at most $\varepsilon\,\mathrm{hfs}(x)$ since hfs is 1-Lipschitz in the intrinsic metric. Moreover, $\mathrm{d}_X(x,p)$ is finite because B_p is included in the path-connected component of X containing p. Therefore, L is a geodesic εhfs-sample of X. Furthermore,

Lemma 23 *For all $\varepsilon \in [0,1]$, the algorithm terminates, and the size its output is within $2^d\,\frac{3+3\varepsilon+2\varepsilon^2}{1-\varepsilon}$ times the size of any geodesic ε*hfs*-sample of X, where d is the doubling dimension of X.*

The proof, given in the full version of this work [51], relies on a simple packing argument. The influence of the doubling dimension d of X is illustrated in Figure 20. In this example, for all point $x \in X$, $\mathrm{hfs}(x)$ is at least half the perimeter of a hole, namely $2 + \frac{2}{2k-1}$. Hence, $P = \{p, p'\}$ and $Q = \{q_1, \cdots, q_k\}$ are geodesic hfs-samples of X. Although Q is hfs-sparse, its size is $|Q| = \frac{k}{2}\,|P|$, where k is of the order of 2^d, as noted in Section 3.4.2.

3.5.3 Computing the homology of a Lipschitz domain

Given a geodesic εhfs-sample L of a Lipschitz planar domain X, a variant of the procedure of Section 3.5.1 can be used to build $D_X(L)$: grow geodesic balls around the points of L at same speed, and report the intersections between the fronts. The homology of $D_X(L)$ gives then the homology of X, by Theorem 15. However, in many practical situations, X is only known through a finite sampling W, which makes it hard to detect the intersections between more than two fronts. In this type of discrete setting, it is relevant to replace the construction of $D_X(L)$ by the ones of $\mathbb{C}^W_X(L)$ and $\mathbb{C}^W_{X,\nu}(L)$, which only require to compare geodesic distances at the points of W. The homology of $D_X(L)$ can then be computed via the persistent homology between $\mathbb{C}^W_X(L)$ and $\mathbb{C}^W_{X,\nu}(L)$.

More precisely, we use simplicial homology with coefficients in a field, which in practice will be $Z/2$ – omitted in our notations. The inclusion map $i : \mathbb{C}_X^W(L) \hookrightarrow \mathbb{C}_{X,\nu}^W(L)$ induces a homomorphism $i* : H_k^\Delta(\mathbb{C}_X^W(L)) \to H_k^\Delta(\mathbb{C}_{X,\nu}^W(L))$. By applying the persistence algorithm [94] to the filtration $\mathbb{C}_X^W(L) \hookrightarrow \mathbb{C}_{X,\nu}^W(L)$, we can compute the rank of i_*. Thus, the goal is to relate rank i_* to $\dim H_k^\Delta(D_X(L))$, the kth Betti number of $D_X(L)$. We know from Theorems 20 and 21 that $\mathbb{C}_X^W(L) \subseteq D_X(L) \subseteq \mathbb{C}_{X,\nu}^W(L)$ under some sampling conditions, which will we assume from now on. The inclusion maps $j : \mathbb{C}_X^W(L) \hookrightarrow D_X(L)$ and $j' : D_X(L) \hookrightarrow \mathbb{C}_{X,\nu}^W(L)$ induce homomorphisms j_*, j'_* on the homology groups, such that $i_* = (j' \circ j)_* = j'_* \circ j_*$. It follows that $\dim H_k^\Delta(D_X(L)) \geq \text{rank } j'_* \geq \text{rank } i_*$, which means that every k-cycle that persists between $\mathbb{C}_X^W(L)$ and $\mathbb{C}_{X,\nu}^W(L)$ is a non-trivial k-cycle of $D_X(L)$. In fact, we even have:

Theorem 24 *Assume that the hypotheses of Theorems 20 and 21 are satisfied, with* $k = \nu$, $L \subseteq W$, *and with* hfs *replaced by* $\min\{\text{hfs}, \text{d}_\text{M}\}$, *where* d_M *is the Euclidean distance to the medial axis* M *of* $R^2 \setminus X$. *Then, the range space of* i_* *is isomorphic to* $H_k^\Delta(D_X(L))$. *Equivalently,* $\text{rank } i_* = \dim H_k^\Delta(D_X(L))$.

This theorem guarantees that the persistent homology between $\mathbb{C}_X^W(L)$ and $\mathbb{C}_{X,\nu}^W(L)$ gives the homology of $D_X(L)$. The bounds on the densities of landmarks and witnesses depend on d_M, which requires that $\text{M} \cap X = \emptyset$. This is true if X has smooth boundaries, but also if ∂X only has corners oriented outwards. The fact that hfs and d_M are both 1-Lipschitz in the intrinsic metric[2] implies that the densities deep inside the domain X can be small, although they have to be large near ∂X.

The proof of Theorem 24 proceeds in two steps: first it shows that j'_* is injective, then it shows that j_* is surjective. It follows from the injectivity of j'_* that $\dim H_k^\Delta(D_X(L)) = \text{rank } j'_*$, which is equal to rank i_* by the surjectivity of j_*.

3.6 Application to sensor networks and simulations

We have implemented the algorithms of Section 3.5 in the context of sensor networks, where the nodes do not have geographic locations, and where the intrinsic

[2] Since d_M is 1-Lipschitz in the Euclidean metric, it is also 1-Lipschitz in the intrinsic metric, because $\text{d}_E \leq \text{d}_X$.

metric is approximated by the shortest path length in the connectivity graph $G = (W, E)$, which is assumed to comply with the *geodesic* unit disk graph model. This means that each node has a geodesic communication range of μ, so that two nodes $w, w' \in W$ are connected in the graph iff $\mathrm{d}_X(w, w') \leq \mu$. All edges have a unit weight, and we denote by d_G the associated graph distance – also called hop-count distance. This geodesic unit disk graph model is the analog of the standard Euclidean unit disk graph model in the intrinsic metric.

Lemma 25 *Assume that W is a geodesic δ-sample of X, with $\delta < \frac{\mu}{2}$. Then, for all nodes $w, w' \in W$, we have:*

$$\frac{\mathrm{d}_X(w,w')}{\mu} \leq \mathrm{d}_G(w,w') \leq \frac{\mathrm{d}_X(w,w')}{\mu}\left(1+\frac{4\delta}{\mu}+\frac{\mu}{\mathrm{d}_X(w,w')}\right)$$

Proof. Let $w, w' \in W$ be two nodes of the graph. We first give an upper bound on d_G. Consider a shortest path ζ from w to w' inside X. We have $|\zeta| = \mathrm{d}_X(w, w')$. Let $0 = t_0 \leq t_1 \leq \cdots \leq t_{m-1} \leq t_m = 1$ be distributed along I in such a way that $\mathrm{d}_X(\zeta(t_i), \zeta(t_{i+1})) = \mu - 2\delta$ for all $i = 0, \cdots, m-2$, while $\mathrm{d}_X(\zeta(t_{m-1}), \zeta(t_m)) \leq \mu - 2\delta$. Clearly, we have $m = \left\lceil \frac{\mathrm{d}_X(w,w')}{\mu-2\delta} \right\rceil$. For all i, let w_i be a point of W closest to $\zeta(t_i)$ in the intrinsic metric. Since W is a geodesic δ-sample of X, we have $w_0 = \zeta(t_0) = w$, $w_m = \zeta(t_m) = w'$, and $\mathrm{d}_X(w_i, \zeta(t_i)) \leq \delta$ for any other i. It follows from the triangle inequality that: $\mathrm{d}_X(w_i, w_{i+1}) \leq \mathrm{d}_X(w_i, \zeta(t_i)) + \mathrm{d}_X(\zeta(t_i), \zeta(t_{i+1})) + \mathrm{d}_X(\zeta(t_{i+1}), w_{i+1}) \leq \mu$. Therefore, $[w_i, w_{i+1}]$ is an edge of the communication graph G, and thus to ζ corresponds a path γ in G. Both ζ and γ connect w to w' and are made of m pieces stitched together. Hence, $\mathrm{d}_G(w, w') \leq m = \left\lceil \frac{\mathrm{d}_X(w,w')}{\mu-2\delta} \right\rceil$, which is bounded from above by:

$$\left\lceil \frac{\mathrm{d}_X(w,w')}{\mu}\left(1+\frac{4\delta}{\mu}\right) \right\rceil \leq \frac{\mathrm{d}_X(w,w')}{\mu}\left(1+\frac{4\delta}{\mu}\right)+1 = \frac{\mathrm{d}_X(w,w')}{\mu}\left(1+\frac{4\delta}{\mu}+\frac{\mu}{\mathrm{d}_X(w,w')}\right).$$

Let us now give a lower bound on d_G. Let γ be any path from w to w' in the communication graph G. For any consecutive nodes w_i, w_{i+1} along the path, we have $\mathrm{d}_X(w_i, w_{i+1}) \leq \mu$ since $[w_i, w_{i+1}]$ is an edge of G. Therefore, by the triangle inequality, γ must have at least $\left\lceil \frac{\mathrm{d}_X(w,w')}{\mu} \right\rceil$ edges. Since this is true for any path γ

from w to w' in G, $\mathrm{d}_G(w,w') \geq \left\lceil \frac{\mathrm{d}_X(w,w')}{\mu} \right\rceil \geq \frac{\mathrm{d}_X(w,w')}{\mu}$. □

Assume now that L is a $\frac{\varepsilon}{1+\varepsilon}$sfs-sparse geodesic εsfs-sample[3] of X. Suppose that $\delta \ll \mu \ll \varepsilon \ll 1$. Given a witness $w \in W$, every landmark $p \in L$ that is not its closest landmark satisfies: $\mathrm{d}_X(w,p) = \Omega(\varepsilon) \gg \mu$, which implies that $\mathrm{d}_G(w,p)$ is an accurate approximation to $\frac{\mathrm{d}_X(w,p)}{\mu}$, by Lemma 25. If now p is the landmark closest to w, then we may as well have $\mathrm{d}_X(w,p) \ll \mu$, but in this case we also have $\mathrm{d}_X(w,p) \ll \mathrm{d}_X(w,q)$ for all $q \in L \setminus \{p\}$, which implies that $\mathrm{d}_G(w,p) < \mathrm{d}_G(w,q)$. As a result, d_G may change the order of the distances between the landmarks and w, but interverted distances must have similar values. In this respect, we can say that d_X is a faithful approximation to d_X, as it is known that the persistent homology of the family of ν-witness complexes is stable under such small perturbations [23].

Homotopy feature size computation. Given a node x, we estimate the geodesic distance of x to its cut-locus, which by Lemma 22 is equal to $\mathrm{hfs}(x)$. In Chapter 2, we have proposed a distributed algorithm for detecting the cut-locus, which works as follows: the node x sends a flood message with initial hop count 1; each node receiving the message forwards it after incrementing the hop count. Thus, every node learns its minimum hop count to the node x. Then, each pair of neighbors check whether their least common ancestor (LCA) is at hop-count distance at least d. If so, then they also check whether their two shortest paths to the LCA contain nodes at least d away from each other (by looking at the $\frac{d}{2}$-ring neighborhoods of the nodes of the paths). Every pair satisfying these conditions is called a cut pair. As proved in Chapter 2, every hole of perimeter greater than d yields a cut pair. Then, every cut node checks its neighbors, and if it has the minimum hop count, then it reports back to x with the hop count value. Thus, x gets a report from one node on each connected component of the cut-locus, and learns the homotopy feature size as the minimum hop value. For a weighted graph, the operation is similar.

Landmark selection and witness complex computation. The landmark selection implements the incremental algorithm of Section 3.5.2 in a distributed manner. A node has two states, *covered* and *uncovered*. A covered node lies inside the

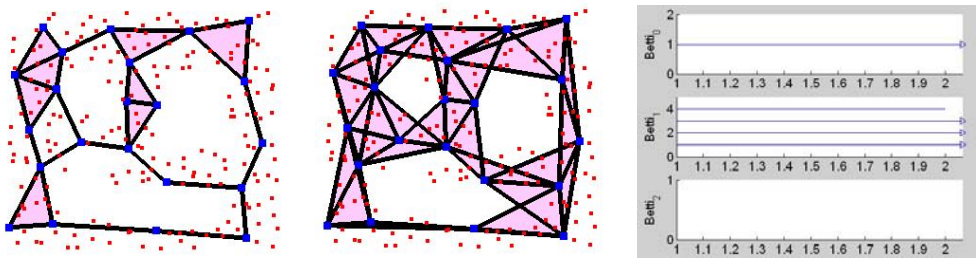
[3] One may as well assume that L is an ε-sparse geodesic ε-sample of X, in a uniform version of the setting.

geodesic ball of some landmark. Initially, all the nodes are uncovered. They wait for different random periods of time, after which they promote themselves to the status of landmark. Each new landmark floods the network, computes its homotopy feature size, and informs all the nodes within its geodesic ball to be covered. Thus, every node eventually becomes covered or a landmark itself.

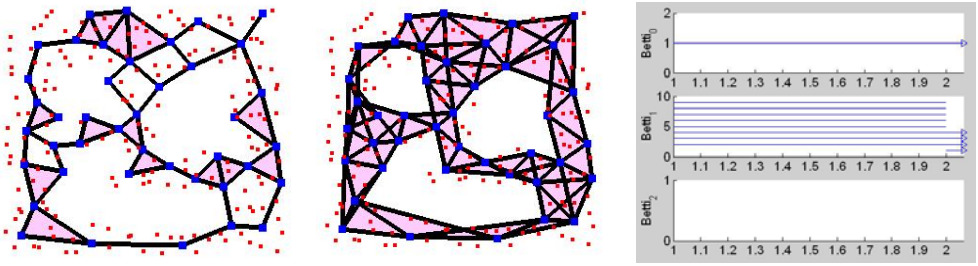
The witness complex is computed in a similar way as in [37]. The selected landmarks flood the network, and every node records its minimum hop counts to them. With this information, it determines which simplices it witnesses. A round of information aggregation collects all the simplices and constructs the witness complex. In a planar setting, where only the Betti numbers β_0 and β_1 are non-zero, we only need to build the 2-skeleton of the witness complex. Therefore, each node may store only its three nearest landmarks, and it may avoid forwarding messages from other landmarks. This reduces the message complexity drastically.

Simulation results and discussion. In this section, we present some simulation results, and we evaluate the dependency of the landmark selection and homology computation on various parameters. The simulations are conducted in a sensor network setting. We consider n sensor nodes randomly distributed in a Lipschitz planar domain. Two nodes within unit Euclidean distance of each other are connected. The resulting average node degree is noted d. The intrinsic metric is approximated by the graph distance in the connectivity network, where each edge can be either unweighted (hop-count distance) or weighted by its Euclidean length (weighted graph distance). Figure 21 shows a typical example, with $\varepsilon = 0.5$ (a) and $\varepsilon = 0.25$ (b). In both cases, only the genuine 3 holes persist and are therefore identified as non-trivial 1-cycles in the geodesic Delaunay triangulation.

- *Node density.* We vary the number of nodes from 217 to 355. The average degree remains the same. The result is shown in Figure 22. Again, the persistent homology between the witness complex and its relaxed version gives the homology of the domain. Thus, only the intrinsic geometry of the domain matters, not the scale of the network, as long as the latter remains dense enough.
- *Landmark density.* Figure 23 shows our results on the same setup as above, with $\varepsilon = 0.85$ (a) and $\varepsilon = 0.15$ (b). In the first case, only two holes are captured, because of the low landmark density. In the second case, three non-genuine

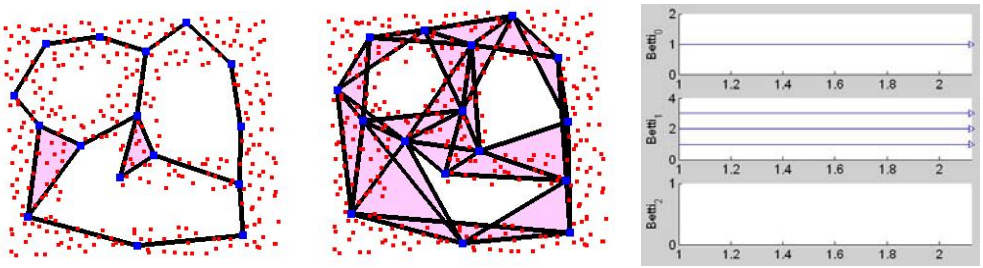


(a) $n = 217$, $d \approx 7.66$, $\varepsilon = 0.5$, $\nu = 2$, weighted graph distance.

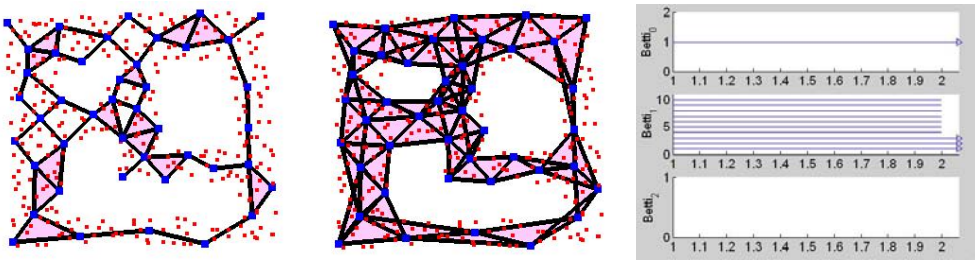


(b) $n = 217$, $d \approx 7.66$, $\varepsilon = 0.25$, $\nu = 2$, weighted graph distance.

Figure 21: From left to right: witness complex, relaxed witness complex, persistence barcodes of the filtration.

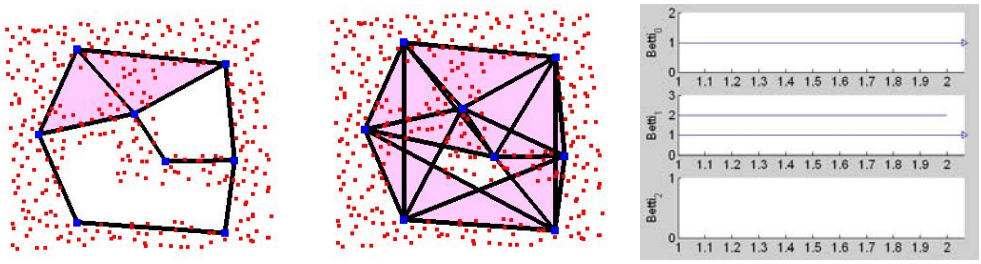


(a) $n = 353$, $d \approx 7.66$, $\varepsilon = 0.5$, $\nu = 2$, weighted graph distance.

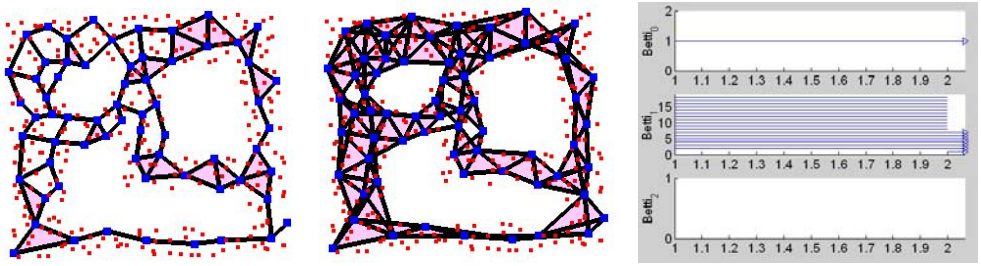


(b) $n = 353$, $d \approx 7.66$, $\varepsilon = 0.25$, $\nu = 2$, weighted graph distance.

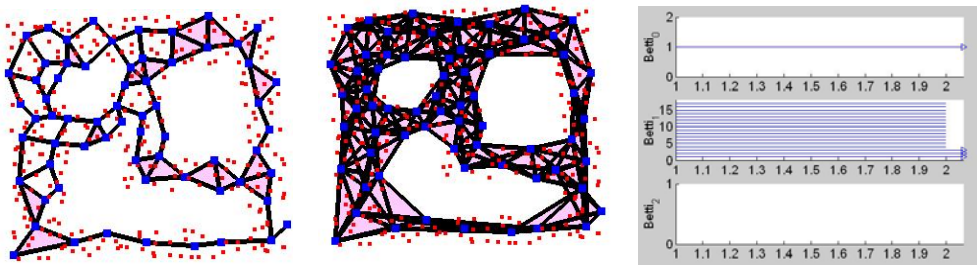
Figure 22: Same setting as above, with same average degree but a higher node density.



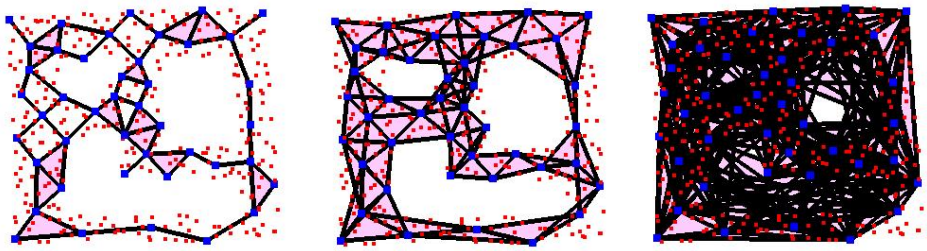
(a) $n = 353$, $d \approx 7.66$, $\varepsilon = 0.85$, $\nu = 2$, weighted graph distance.



(b) $n = 353$, $d \approx 7.66$, $\varepsilon = 0.15$, $\nu = 2$, weighted graph distance.



(c) $n = 353$, $d \approx 7.66$, $\varepsilon = 0.15$, $\nu = 4$, weighted graph distance.



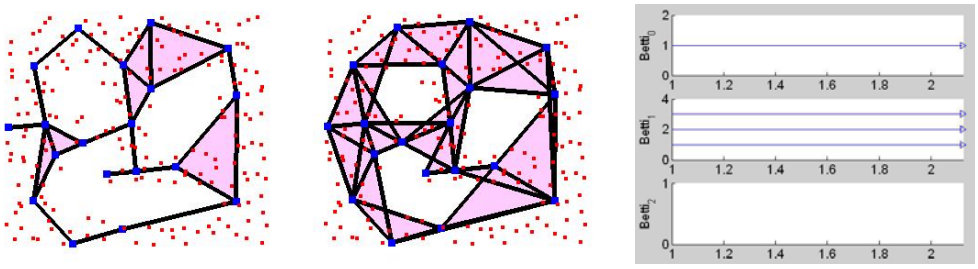
(d) $n = 353$, $d \approx 7.66$, $\varepsilon = 0.25$, weighted graph distance. Left: witness complex; Middle: $\nu = 2$; Right: $\nu = 11$.

Figure 23: Same setting as above, effect of varying parameter ν, versus landmark density.

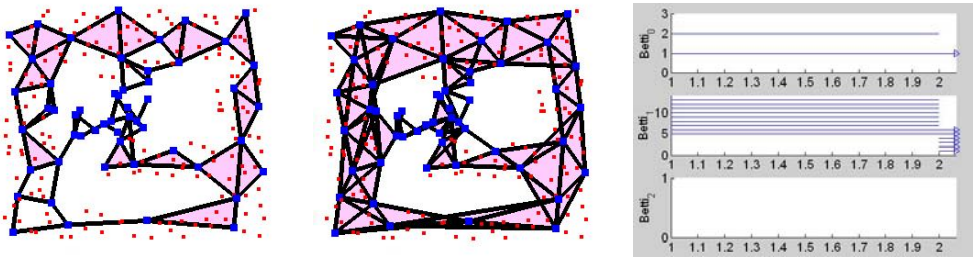
holes are not destroyed in the relaxed witness complex, because the value of the relaxation parameter ν is too small given the relatively low node density. Increasing ν from 2 to 4 produces the correct answer (c). But setting ν to too high a value ($\nu = 11$, $\varepsilon = 0.25$) destroys some of the genuine holes (d).

Throughout our experiments, the algorithm produced correct results with small values of ν ($\nu \leq 4$), provided that the nodes and landmarks sets were reasonably dense. This demonstrates the practicality of our approach, despite the large theoretical bounds stated in Theorems 20, 21 and 24.

- *Weighted graph distance vs. hop-count distance.* Since the hop-count distance is a poor approximation to the geodesic distance, the range of values of ε that work fine with it is reduced. In Figure 24 for instance, the scheme works well with $\varepsilon = 0.5$, but not with $\varepsilon = 0.25$, in contrast with the results of Figure 21.
- *Packing strategy.* Figure 25 shows some of our sampling results. It appears that different packing strategies can produce samples of very different sizes, as predicted by Lemma 23. Maximizing the ratio $\frac{\mathrm{d}_X(q,L)}{\mathrm{hfs}(q)}$ at each iteration seems to be a very effective strategy in practice, but its is also time-consuming.

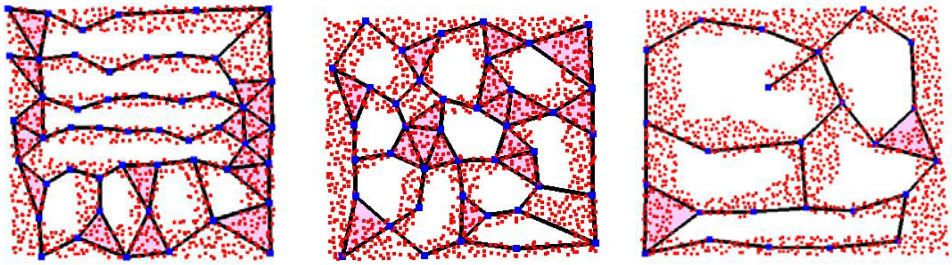


(a) $n = 217$, $d \approx 7.66$, $\varepsilon = 0.5$, $\nu = 2$, hop-count distance.

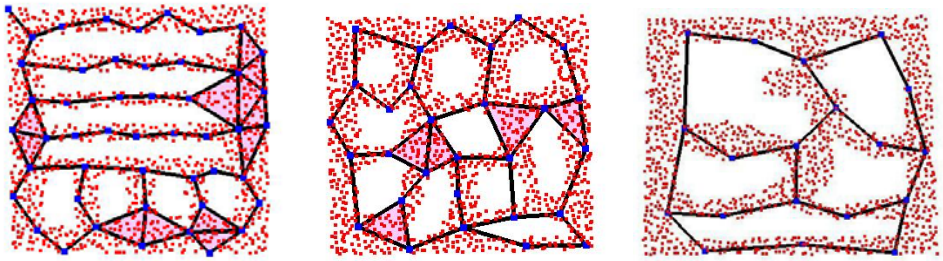


(b) $n = 217$, $d \approx 7.66$, $\varepsilon = 0.25$, $\nu = 2$, hop-count distance.

Figure 24: Same setting as above, with the weighted graph distance replaced by the hop-count distance.



(a) $\varepsilon = \frac{1}{3}$, random landmark selection outside $\bigcup_{p \in L} \overline{B}_p$.



(b) $\varepsilon = \frac{1}{3}$, insertion of node q that maximizes $\frac{d_X(q,L)}{\mathrm{hfs}(q)}$ outside $\bigcup_{p \in L} \overline{B}_p$.

Figure 25: Landmark sets obtained by two different packing strategies, and their geodesic witness complexes.

3.7 Conclusion

We have introduced a new quantity, called the homotopy feature size, and showed that it is well-suited for the sampling and analysis of Lipschitz domains in the plane. In particular, given a domain X and a landmark set L that is sufficiently densely sampled from X, the bound on the density depending on the homotopy feature size of X, we have proved that the geodesic Delaunay triangulation of L is homotopy equivalent to X. The homotopy feature size depends essentially on the global topology of X, and it is rather insensitive to the local geometry. As a result, it enables to have very sparse sets of landmarks, which makes it a convenient theoretical tool for geometric data analysis.

With more practical applications in mind, we have focused on the geodesic

witness complex and its relaxed version, proving that these two complexes sandwich the geodesic Delaunay triangulation under some conditions. As an application, we have shown that it is possible to estimate the homology of a Lipschitz planar domain X from a finite set of landmarks L without actually building $D_X(L)$ explicitly, by constructing $\mathbb{C}^W_X(L)$ and $\mathbb{C}^W_{X,\nu}(L)$ and computing their persistent homology. To give theoretical guarantees to this approach, we proved that the persistent homology between $\mathbb{C}^W_X(L)$ and $\mathbb{C}^W_{X,\nu}(L)$ coincides with the homology of $D_X(L)$, yet under some fairly stringent sampling conditions. Our practical experiments in the context of sensor networks suggest that milder conditions should be sufficient.

This work can be generalized in several ways. In a near future, we intend to look at possible extensions for bounded domains in higher-dimensional Euclidean spaces, with applications in robotics and geometric data analysis. Also, it would be relevant to generate homology bases whose elements isolate the various holes of X. There exists some work along this line, but for a slightly different context [42]. Finally, in order to make the approach fully practical, it would be necessary to devise distributed variants of the procedures that build the simplicial complexes and compute the persistent homology. Whether such variants exist is still an open question at this time.

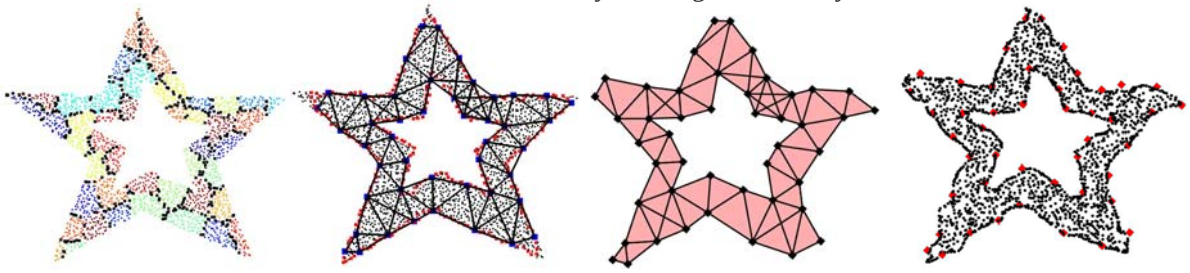
Chapter 4

Discovery of Sensor Network Layout using Connectivity Information

4.1 Introduction

In the chapter, we present our method to recover the global network layout and the core of our localization algorithm. We assume the sensor nodes are embedded in a geometric region or on a terrain possibly with holes (corresponding to obstacles). The nodes nearby can directly communicate with each other but far away nodes can not. We do not use anything beyond the network connectivity information and do not assume the neighbors can measure their inter-distances, although such information can certainly help and further improve the localization accuracy.

The algorithm can be explained in four steps as shown in Figure 26. Suppose the network boundaries (both the outer boundary and inner hole boundaries) have been discovered (say with any of the algorithms in [39, 40, 44, 45, 68, 89]). 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 26(i)). 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) (Figure 26(ii)). We can prove that the Delaunay complex is rigid when landmarks are sufficiently



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

Figure 26: 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.

dense in the case of a continuous domain (i.e., corresponds to the sensor density approaching infinity).

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 any more! 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 thesis we make this intuition rigorous. We prove in the case of a continuous geometric domain when the landmarks are sufficiently dense (with respect to the local geometric complexity), the induced Delaunay graph is rigid. In addition, 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 the way to embed them relative to each other remove a major hurdle towards 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 balance the error distribution (Figure 26 (iii)). In practice, to achieve lower

overhead we can also only embed the boundary nodes using trilateration and then perform a rubberband relaxation scheme on the remaining nodes. This is actually what we did in our simulations.

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.

4.2.1 Medial axis, local feature size and r-sample

We consider a geometric region R with obstacles inside. The boundary ∂R consists of the outer boundary and boundaries of inner holes. For any two points $p, q \in R$, we denote by $|pq|$ their Euclidean distance and $d(p,q)$ the *geodesic* distance between them inside 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 chapter 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 26 *The* medial axis *of R is the closure of the collection of points, with at least two closest points on the boundary ∂R.*

The medial axis of ∂R consists of two components, one part inside R, called the *inner medial axis,* and the other part outside R, called the *outer medial axis*. See Figure 27. In this chapter 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 27 *The inner medial axis of 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 ∂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 ∂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 R, which determines the sampling density. An example is shown in Figure 27.

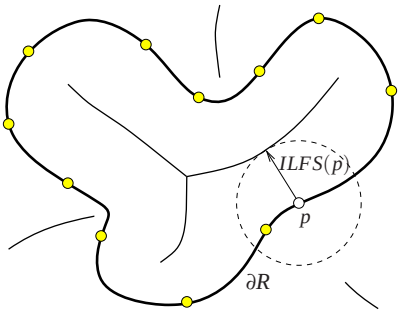


Figure 27: The region R's boundary is shown in dark curves. The medial axis and landmarks selected on the boundaries. Point $p \in \partial R$ has a landmark within distance $ILFS(p)$.

Definition 28 *The* inner local feature size *of a point $p \in \partial R$, 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 $p \in \partial R$, 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 29 *An r-sample of the boundary ∂R is a subset of points S on ∂R such that for any point $p \in \partial R$, 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.

4.2.2 Landmark Voronoi diagram and combinatorial Delaunay graph

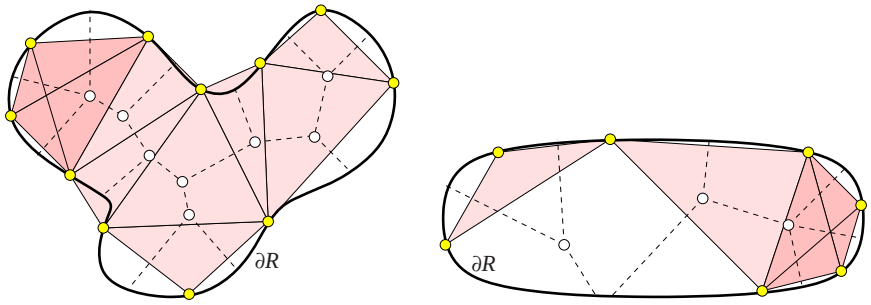
We take some points in R and denote them as *landmarks S*. Construct the *landmark Voronoi diagram* $V(S)$ as in [37]. Essentially each point in R identifies the closest landmark in terms of geodesic distance. The *Voronoi cell* of a landmark u, denoted as $V(u)$, includes all the points that have u as a closest landmark:

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

Each Voronoi cell is a connected region in R. The union of Voronoi cells covers the entire region 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 ∂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 28 for some examples.

We state some immediate observations about the Voronoi diagram and the corresponding combinatorial Delaunay graph below.

Observation 30 *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 $D(S)$.*



(i) (ii)

Figure 28: (i) The Voronoi graph (shown in dashed lines) and the Delaunay graph/complex for a set of landmarks that form an r-sample with $r < 1$. (ii) When the set of landmarks is not an r-sample (with $r < 1$), the combinatorial Delaunay graph may be non-rigid.

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 28 (i). We will capture these high-order features by defining the Delaunay complex in the notion of abstract simplicial complex [33]. 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 $\dim\alpha = |\alpha| - 1$, 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 *Čech complex* of the Voronoi cells, defined below.

Definition 31 *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 28 (ii) 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 chapter, 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 $\sigma = \text{conv}\, S$. The dimension of σ is $\dim \sigma = k$. Figure 29 shows 0, 1, 2, 3-simplex in $\mathbb{R}^3$.

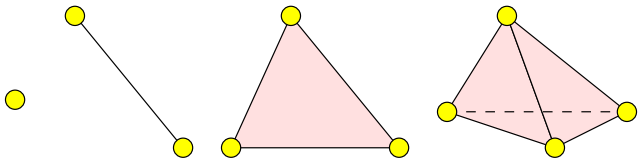


Figure 29: 0, 1, 2, 3-simplex in $\mathbb{R}^3$.

The convex hull of any subset $T \subseteq S$ is also a simplex. It is a subset of $\text{conv}\, S$ 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 $\text{conv}\varphi(\alpha) \in K$ [33]. 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 chapter, 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.

4.2.3 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 ∂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 28 (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 R. In this subsection we assume that the landmarks are selected according to the landmark selection criterion mentioned above.

We prove an important Lemma about the inner local feature size first. This Lemma and its proof are motivated by [3] and will be useful for other proofs in this subsection.

Lemma 32 *Given a disk B containing at least two points on ∂R, for each connected component of $B \cap R$, either it contains a point on the inner medial axis, or its intersection with ∂R is connected.*

Proof. We take one connected component C of $B \cap R$ and assume that it does not contain a point on the inner medial axis and intersects ∂R in two or more connected pieces. Now we take a point u in C but u is not on ∂R. Now take u's closest point

on $C \cap \partial R$. 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 $C \cap \partial R$. We take u's closest point on a different piece of $C \cap \partial R$, denoted as q. See Figure 30.

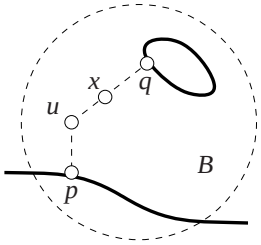


Figure 30: Each connected component of $B \cap R$ either contains a point on the inner medial axis or its intersection with ∂R is connected.

Now as we move a point x from u to q along the geodesic path between u and q, x's closest point on $C \cap \partial R$ 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. □

Observation 33 *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 34 *For any two adjacent landmarks u, v on the same boundary cycle, there must be a Voronoi vertex inside R whose closest landmarks include u, v.*

Proof. We take two adjacent landmarks u, v and consider the set of points in 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 31.

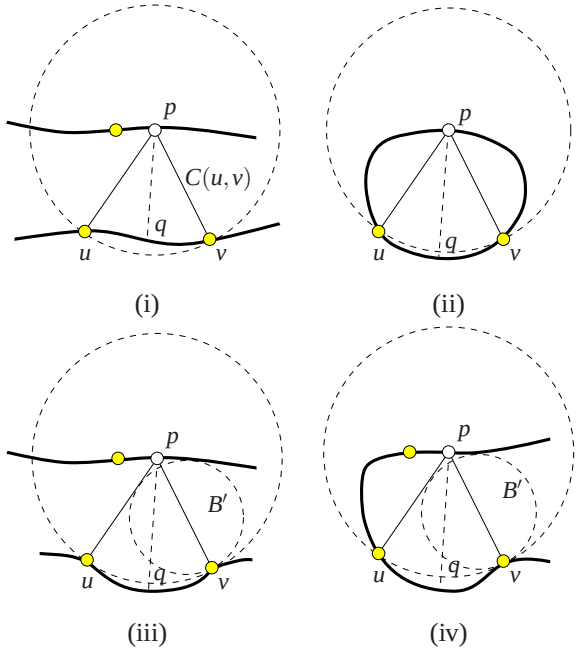


Figure 31: u, v are two adjacent landmarks. The point p on the boundary has its closest landmarks as u, v. (i)-(iv) four possible cases.

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 $B_{r^-}(p) \cap R$ that contains the curve

$C(u,v)$, denoted by F. By Lemma 32, if F does not contain a point on the inner medial axis, then its intersection with the boundary ∂R is connected. Now we do a case analysis depending on how the boundary curve goes through u and v. In Figure 31 (i) & (ii), the ε-neighborhood of the boundary at u, v also intersects $B_{r^-}(p) \cap R$. In (i), $F \cap \partial R$ 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 $B_{r^-}(p) \cap R$ (meaning that $B_{r^-}(p)$ does not contain any ε-neighborhood of v, see Figure 31 (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 34 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 35 *If there is a continuous curve C that connects two points on the boundary ∂R such that C does not contain any point on Voronoi edges, then C cuts off a topological 1-disk*[1] *of ∂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 32(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 ∂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

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

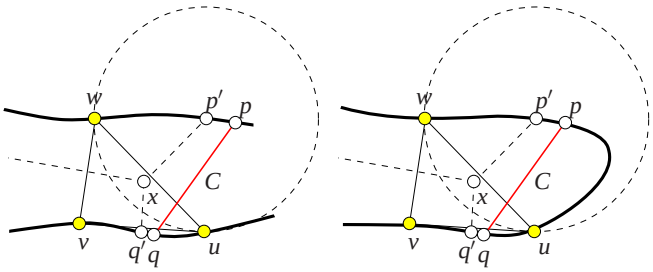


Figure 32: (i) C is inside the Voronoi cell of landmark u to the right of C. (ii) the curve C cuts off a segment of ∂R with no other landmark inside.

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 32 that the component of $B_r(p') \cap R$ containing p' intersects ∂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 $B_r(p') \cap R$ containing p' intersects ∂R in a connected piece, this intersection is a continuous segment between u and w on ∂R, completely inside $B_r(p')$, by using the same argument as in the previous lemma; see Figure 32 (ii). In this case, the curve C cuts off a segment of ∂R with at most one landmark inside. The claim is true. □

Corollary 36 *The Voronoi graph $V(S)$ is connected.*

Proof. This follows immediately from Lemma 34 and Lemma 35, although Lemma 35 is stronger. Specifically if $V(S)$ is not connected, we are able to find a curve C that cuts 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 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 37 (Laman condition [69]) *A graph G with n vertices is generically rigid* [2] *in 2 dimensions if and only if it contains a Laman graph on n vertices.*

Theorem 38 *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 ∂R and we ignore those. By Corollary 36 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

[2] *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.*

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 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 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 $\mathrm{conv}\,\varphi(\sigma) \in \mathbb{R}^2$. 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 [10, 60]

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 chapter, 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 33(i).

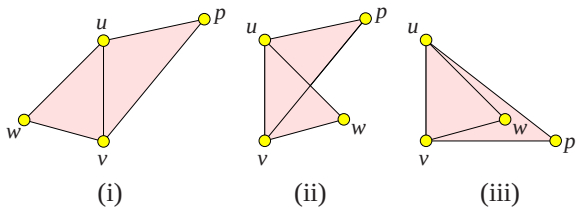


Figure 33: Two Delaunay triangles $\triangle uvw$ and $\triangle uvp$ sharing an edge. (i) is the only valid embedding with the two triangles not sharing any interior points.

Otherwise, w and p are embedded on the same side of uv. Then either w is inside $\triangle uvp$ (as in Figure 33 (iii)), or p is inside $\triangle uvw$, or two edges intersect at a non-vertex point (as in Figure 33 (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 39 *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.*

4.2.4 Topological equivalence

Our sampling condition aims to capture the geometric complexity of the region R. A related question may ask whether the constructed geodesic Delaunay complex is homotopy equivalent[3] to the region 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 R. Our current sampling condition, unfortunately, can not guarantee the homotopy equivalence. A bad example is shown in Figure 34. 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 chapter 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 R is the topic of Chapter 3 [50], 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[4], the homotopy equivalence is established. Thus we can check locally whether the conditions are satisfied.

[3]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.

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

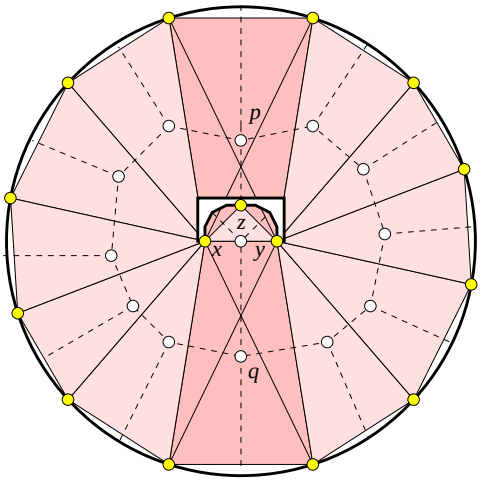


Figure 34: A nasty example with no valid embedding of the Delaunay complex.

Theorem 40 *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* R.

Proof. As the combinatorial Delaunay complex is the Cěch complex of the Voronoi cells, the theorem follows immediately from the Cěch Theorem [18]. Recall the definition of the Cěch complex. Given a collection of sets $U = \{V(u), \forall u \in S\}$, the *Cěch complex* is the abstract simplicial complex whose k-simplices correspond to nonempty intersections of $k+1$ distinct elements of 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 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 34 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.

4.3 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 in Chapter 2 [89]. 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 26(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.

4.3.1 Select landmarks

We use a distributed boundary detection algorithm that identifies nodes on both outer and inner boundaries and connects them into boundary cycles [89]. 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 [19]. The boundary nodes flood inward at roughly the same time [34, 48]. 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 [19, 93]. The idea is to take the nodes with two or more closest intervals on the network boundary [93]. 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.

4.3.2 Compute Voronoi diagram and combinatorial Delaunay complex

The landmark Voronoi diagram is computed in a distributed way as in [37]. 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 41 *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 26 (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 42 *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 [22,29]. This is explored in Chapter 3 [50].

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.

4.3.3 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 45), 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 [43, 62, 63, 66, 78] 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 [66, 78].

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 $p_i p_j$. 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 [9, 21]. 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.

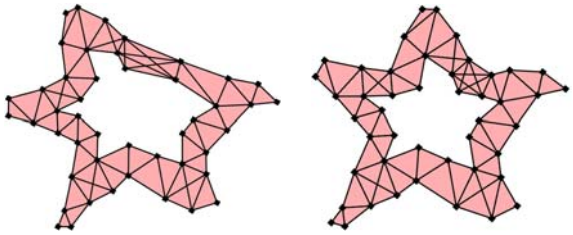


Figure 35: Left: before the mass-spring relaxation algorithm is applied; Right: after mass-spring relaxation.

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

4.3.4 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 [82]) 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.

4.4 Simulations

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

4.4.1 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.

4.4.2 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. [86] 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 [47, 80], 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 4 (ii) and Figure 37.

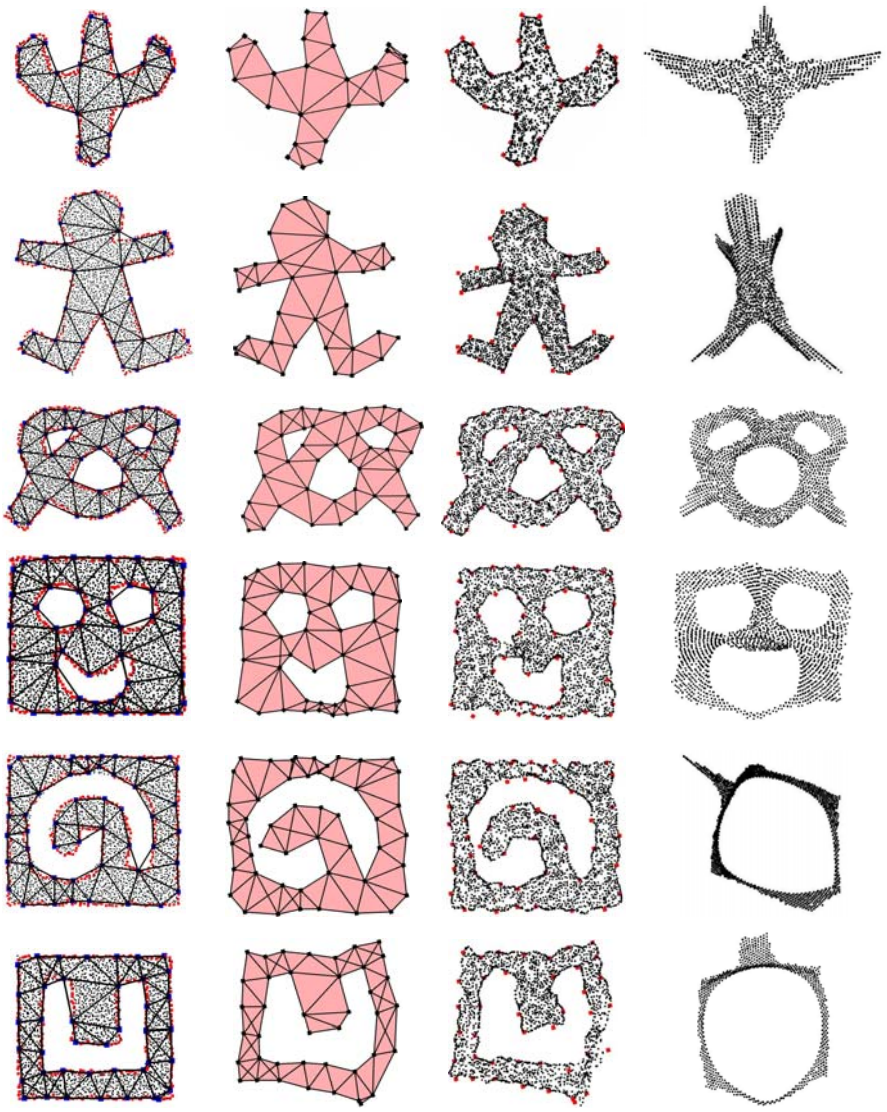
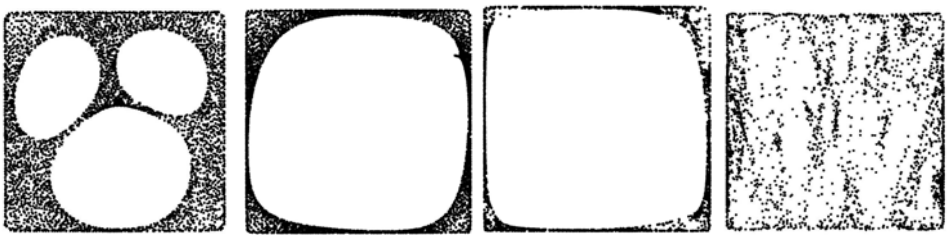


Figure 36: From left to right, we have: (i) the true sensor locations and extracted combinatorial Delaunay complex; (ii) embedding of the combinatorial Delaunay complex; (iii) localization of all nodes by our algorithm; (iv) the results produced by MDS on all nodes in the network. The connectivity network 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)

Figure 37: Rubberband algorithm results for (i) face (ii) spiral in a box (iii) square with a concave hole (iv) U shape.

4.4.3 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.

4.4.3.1 Influence of network shapes

We applied our algorithm to a number of networks with different layouts, or "shapes". We observed that the performance 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 36.

Figure 36 (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 36 (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[5]. In all cases

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

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 1.

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 1: Average location error, scaled by communication range.

4.4.3.2 Influence of network communication models

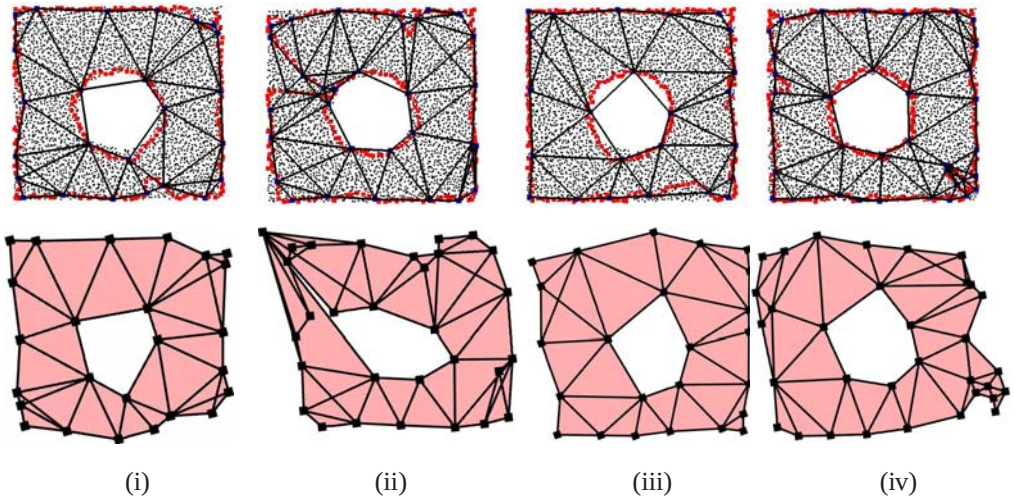


Figure 38: Embedding the landmarks under challenging network conditions. The first row shows the ground truth; the second row our embedding of the landmark nodes. From left to right the models depicted are (i) 3443 nodes, avg. degree 10.66. only keep α edges and delete (1-α) edges randomly. α= 0.9. (ii) 3443 nodes, avg. degree 11.95. α = 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. α = 0.8 (iv) 3443 nodes, avg. degree 7.57. α = 0.6.

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 38. Figure 38 (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 $(1-d)/(1-\alpha)$. If $d < \alpha$, there must be an edge between them. $\alpha = 0.8$ 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.

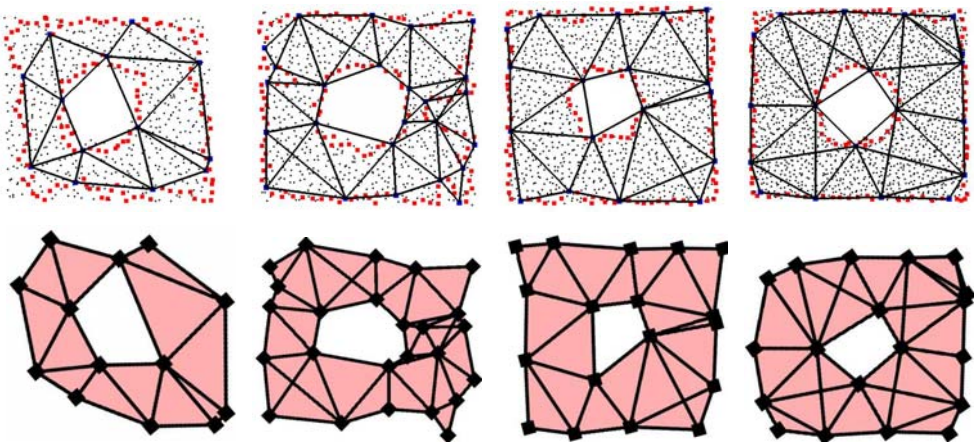
4.4.3.3 Influence of node density

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 39 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.

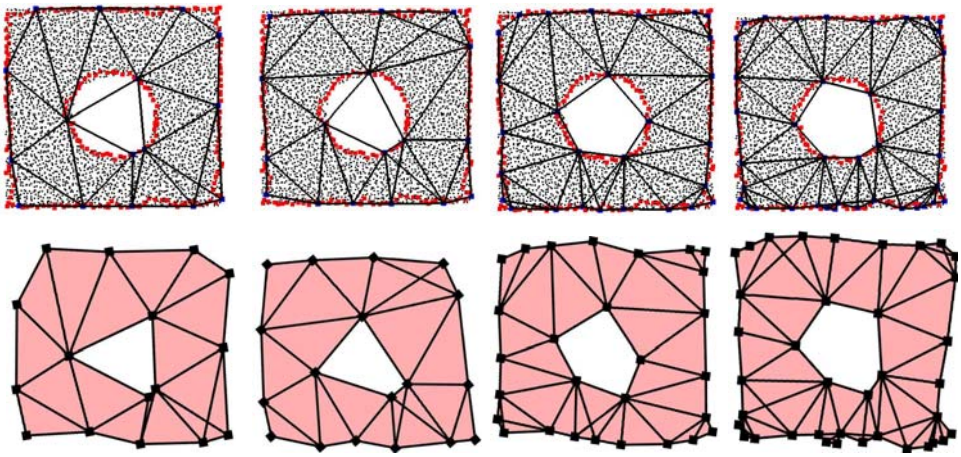
4.4.3.4 Influence of landmark density

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



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

Figure 39: 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.

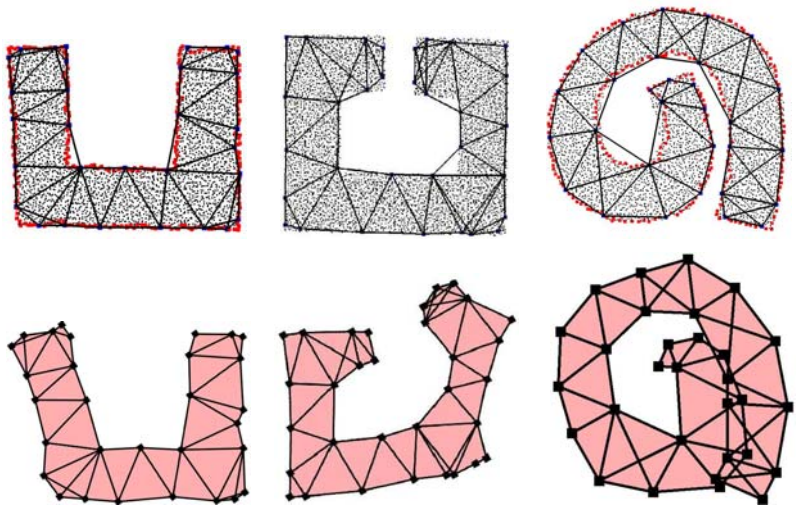


(i) (ii) (iii) (iv)

Figure 40: 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 even more

may allow for a slightly better embedding 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 40, 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.

4.4.3.5 Error accumulation



(i) (ii) (iii)

Figure 41: 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.

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 41. In these examples, the embedded shape is distorted and may have self-overlap (as in example (iii)), due to error accumulation.

4.4.4 Further discussion

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 [88], 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.

4.5 Conclusion

In this chapter, 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 chapter to be helpful in recovering the network geometry.

Chapter 5

Sensor Network Localization with Incremental Delaunay Refinement

5.1 Introduction

As we have discussed in Chapter 4, the location of sensor nodes is an indispensable component for both network operation and sensor data integrity. We proposed an algorithm for landmark selection to guarantee that the generated combinatorial Delaunay complex is globally rigid and admits a unique realization in the plane. This leads to an algorithm to put together the Delaunay triangles in an incremental manner which helps localize the rest of the nodes in the network.

One property of our algorithm in Chapter 4 is to select landmarks first using a boundary detection algorithm [39,40,44,45,68,89] to identify the network boundary nodes and then selects the landmarks to be a γ-sample with $\gamma < 1$. Specifically, every boundary node has a landmark within its inner local feature size, defined as the distance to the closest node on the medial axis (which is the collection of nodes with two or more closest nodes on the boundary). The dependency on the boundary detection algorithm puts limitations on the applicability of the localization algorithm as all known boundary detection algorithms do not work well in the case of extremely low density networks. Examples of some of these cases were shown in Chapter 4.

The main contribution in this chapter is an incremental landmark selection algorithm that does not assume knowledge of the network boundary. In particular, we start with no knowledge of the network topology (whether there are holes or how many there are, etc.) and develop local conditions to test whether a node should be included as a new landmark. The landmarks selected naturally adapt to the local geometry of the network, with a higher density of landmark nodes selected in regions with more detailed and complex features. This new landmark selection algorithm greatly enhances the robustness of our algorithm in cases of extremely sparse or even non-rigid networks, or networks with very complicated shapes that are challenging for boundary detection algorithms. We are not aware of any other localization algorithms using only connectivity information with comparable performance. We demonstrate the improved performance of our algorithm in various network settings in the simulation section.

5.2 Localization by Delaunay complex

In this section we use a continuous setting to go though the framework of network localization by the combinatorial Delaunay complex and provide the theoretical foundation of the incremental Delaunay refinement algorithm. The algorithm implementation in the network setting is elaborated on in the next section.

We firstly recall some assumptions and concepts which we have described in Chapter 4. The sensor field is assumed to be a continuous domain $R \in \mathbb{R}^2$ with perhaps some interior holes. For any two points $p, q \in R$, we denote by $|pq|$ their Euclidean distance and $d(p,q)$ the *geodesic* distance (shortest path distance) between them inside R. The geodesic distance is an analog of the minimum hop count distance in the discrete setting. 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.

The boundary of R is denoted as ∂R and may have multiple cycles. The *inner medial axis* of R is the collection of points in R that have two or more closest points on the boundary ∂R. The *inner local feature size* of a point $p \in \partial R$, denoted by $ILFS(p)$, is the distance from p to the inner medial axis of R. A set of landmarks

L on the boundary ∂R is called a γ-sample[1] if for any point $p \in \partial R$, there is at least one landmark within distance $\gamma \cdot ILFS(p)$ from p.

Suppose L is a set of landmarks on the domain boundary ∂R, the *Voronoi cell* of a landmark u, denoted as $V(u)$, includes all the points that have u as a closest landmark:

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

The collection of Voronoi cells is denoted as the landmark Voronoi diagram $V(L)$ for the set L of landmarks. A point is called a *Voronoi vertex* if it has equal distance to at least three landmarks. The Voronoi vertices inside R are called the *inner Voronoi vertices*. A ball $B_r(p)$ centered at an inner Voronoi vertex p with radius r equivalent to the distance from p to the closest landmarks is called a *Voronoi ball*.

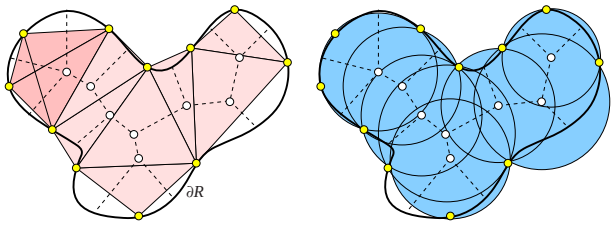


Figure 42: Left: The Voronoi graph (shown in dashed lines) and the Delaunay complex for a set of landmarks on the boundary ∂R. The Delaunay simplices (vertices, edges, triangles, tetrahedrons) are shaded. Right: The union of Voronoi balls approximately covers the domain R.

The *combinatorial Delaunay complex* of the landmarks L, denoted by $DC(L)$, is the collection of sets

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

In other words, a set $\alpha \subseteq L$ is a Delaunay simplex if the intersection of the Voronoi cells of landmarks of α is non-empty. The Delaunay complex has naturally 0-dimensional simplices such as the landmarks, 1-dimensional simplices such as Delaunay edges, and 2-dimensional simplices such as Delaunay triangles, and possibly

[1] Notice that the definition of γ-sample is different from the typical definition in geometric processing [3, 4] where the local feature size is used. The medial axis for a domain R has two parts, one inside R and one outside R. For our setting we do not have access to the part of the medial axis outside of R and we can only use the inner local feature size.

higher order simplices such as tetrahedrons. See Figure 42 for an example.

5.2.1 γ-sample, rigidity and coverage

In our previous work in Chapter 4 [70], we show a framework for network localization by embedding the Delaunay complex $DC(L)$ extracted from the network connectivity. The main result in Chapter 4 is a proof that when the landmarks are selected as a γ-sample of the domain R with $\gamma < 1$, the Delaunay complex $DC(L)$ is globally rigid and thus admits a unique realization in the plane. This establishes the foundation of the localization algorithm as we can now embed the Delaunay complex incrementally and then localize the entire network with the Delaunay complex as a structural skeleton.

For localization, we also want that the Delaunay complex provides good 'coverage' of the sensor field in the sense that every node is not very far from the Delaunay complex, so that the Delaunay complex faithfully represents the network shape. In particular, we take B to denote the union of all the Voronoi balls, and U the shape of the union of these balls. As we will prove later, the γ-sample guarantees that the union of Voronoi balls is a good approximation of R and the approximation is improved as the density of landmarks increases. See Figure 42 (ii) for an example. Rigorously, we define that the Delaunay complex $DC(L)$ δ-*covers* R if every point $x \in R$ will be within distance $(1+\delta)\cdot r$ from the center p of a Voronoi ball $B_r(p)$, where r is the radius of this Voronoi ball.

Using the union of the Voronoi balls to approximate the shape R was initially proposed in geometric processing and computer graphics [4]. It has been shown that the errors in the position and normal of the surface of U with R is bounded everywhere, given sufficiently dense samples on the ∂R. However, we cannot directly apply the results in [4] as there are a couple of differences with our setting. First, the metric we are working with is the geodesic shortest path metric, instead of the Euclidean metric used in [4]. In addition, as we only have sensors in the interior of R, we do not have access to the part of medial axis that is outside R and we are only able to use the inner local feature size to define the γ-sample.

Before we prove the coverage theorem, let us first understand the boundary of

the union of balls U. The boundary of U contains some circular arcs from the balls in B. We first characterize what arc can possibly stay on the boundary of U. Each Voronoi edge in $V(L)$ has two endpoints, being either a Voronoi vertex or a point on the boundary ∂R. A Voronoi edge with two Voronoi vertex endpoints is called an *inner edge*. A Voronoi edge with two endpoints on the boundary is called an *outer edge*. A Voronoi edge with both a Voronoi vertex and a point on the boundary is called a *mixed edge*. For each Voronoi ball B, the three landmarks partition its boundary ∂B into three circular arcs. We label the arc between two landmarks u, v with the label of the Voronoi edge of u, v as either inner, outer, or mixed. We now claim that only mixed arcs can possibly appear on ∂U. First realize that the interior points of an inner arc cannot stay on the boundary of U, since the arc is enclosed inside the union of the two Voronoi balls that go through u, v. Second if we choose $\gamma < 1$, then the Voronoi diagram inside R is connected as proved in Corollary 35 in Chapter 4. Thus there cannot be an outer edge in $V(L)$, since this edge will be disconnected from the rest of the Voronoi diagram. Now for a Voronoi ball $B_r(p)$ with a mixed edge between landmarks u, v we define a *pie* as the set of points in R bounded by the boundary segment between u, v and the shortest paths from p to u, v. Only the points inside a pie with a mixed arc can possibly stay outside U. See Figure 43. Notice that in the case of a degeneracy, a Voronoi ball can have four or more landmarks. The classification of edges and the proof later are the same in that case.

Lemma 43 (Lipschitz continuity) *The inner local feature size of any shape $R \subseteq \mathbb{R}^2$ is 1-Lipschitz: $ILFS(x) \leq ILFS(y) + d(x,y)$ for any $x, y \in \mathbb{R}^2$.*

Proof. The proof follows from triangle inequality. Suppose that point p is the closest point of y on the inner medial axis of R. Then $d(p,y) = ILFS(x)$. Thus, $ILFS(x) \leq d(p,x) \leq d(p,y) + d(x,y) = ILFS(y) + d(x,y)$. □

Theorem 44 *For a connected region $R \subseteq \mathbb{R}^2$, we select landmarks L as a γ-sample on the region boundary ∂R with $\gamma < 1$. Then the Delaunay complex δ-covers R, with $\delta = 2\gamma/(1-\gamma)$.*

Proof.

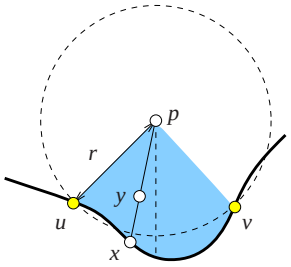


Figure 43: Any x is within distance $\delta \cdot r$ from a Voronoi ball. A pie between a mixed arc $\hat{uv}$ is shown in shade.

We first prove the claim for points on ∂R .

Consider a point x on ∂R in between two landmarks u, v, as shown in Figure 43. Lemma 33 in Chapter 4 says that there is a Voronoi vertex p with u, v as two closest landmarks and the Voronoi edge with respect to u, v is a mixed edge. Without loss of generality we assume that x's closest landmark is u. By the γ-sample property, $d(u,x) \leq \gamma \cdot ILFS(x)$.

Now we assume by contradiction that $d(p,x) > (1+\delta)r$. Thus $\gamma \cdot IFLS(x) \geq d(u,x) \geq d(p,x) - d(p,u) > (1+\delta)r - r = \delta r$ by the triangle inequality. Thus $ILFS(x) > \delta r/\gamma$.

We also know that the inner local feature size is a 1-Lipschitz function with proof in Lemma 43. That is, $ILFS(x) \leq ILFS(u) + d(u,x)$. As we know that the Voronoi ball $B_r(p)$ touches three landmarks and contains at least one point on the medial axis in R , $ILFS(u) \leq 2r$. Thus we have, $ILFS(x) \leq 2r + \gamma \cdot ILFS(x)$. Combining the inequalities, we have $\delta r/\gamma < IFLS(x) \leq 2r/(1-\gamma)$. That gives us $\delta < 2\gamma/(1-\gamma)$, a contradiction.

If the claim is true for all points on ∂R , it is true for all points in R . Suppose otherwise, then there is a point y in the interior of R that is not δ-covered. y can only possibly stay inside a pie, as shown in Figure 43. Then there must be another point $x \in \partial R$ such that y stays on the geodesic shortest path from p to x. Thus y is covered by $B_r(p)$, the same Voronoi ball that covers x. □

5.2.2 Landmark selection for both rigidity and coverage

Based on the previous discussion, there are two desirable criteria, namely, global rigidity and coverage, for the final Delaunay complex. In this subsection we investigate *local* conditions for landmark selection to guarantee both rigidity and good coverage of the induced Delaunay complex:

1. **Local Voronoi edge connectivity:** The Voronoi edges for each landmark u form a connected set.
2. **Local Voronoi ball coverage:** Each node x inside a Voronoi cell $V(u)$ is δ-covered by a Voronoi ball $B_r(p)$, where p is a Voronoi vertex with landmark u.

We first show that if both conditions are satisfied for a set of landmarks L, then the Delaunay complex $DC(L)$ satisfies both the global rigidity and coverage property. This is relatively straightforward. After this, we examine how to design a landmark selection algorithm to meet these conditions.

5.2.2.1 Rigidity of the Delaunay complex

When the local Voronoi edge connectivity condition is met, we argue that the Delaunay complex is globally rigid. To do that, we will make use of a theorem proved in Chapter 4:

Theorem 45 *[Global Rigidity] If $V(L)$ is connected inside R, the Delaunay complex $DC(L)$ is globally rigid.*

The local Voronoi edge connectivity immediately implies the global Voronoi edge connectivity. If otherwise, there must be one landmark whose Voronoi edges have two or more connected components, since the union of all the Voronoi cells is R. Thus the local Voronoi edge connectivity condition implies the global rigidity of $DC(L)$.

5.2.2.2 Coverage of the Delaunay complex

If the local Voronoi ball coverage condition is met for every Voronoi cell, then the coverage property of the Delaunay complex follows directly.

5.2.2.3 Incremental Delaunay refinement algorithm

Since both conditions can be tested locally, we naturally have the following incremental landmark selection algorithm: for each Voronoi cell $V(u)$,

1. If the first condition is not met, the Voronoi edges with u have two or more connected components. Since each Voronoi edge has either a Voronoi vertex or a point on ∂R as endpoints, we select, among all the endpoints of Voronoi edges of u on ∂R, the one that is *furthest* from u as a new landmark.

2. If the first condition is met, we check the second condition. Among all the points that violate the local Voronoi ball coverage condition, we select the one that is least covered as a new landmark: $\max_x \min_{B_r(p)}\{\delta' | d(x,p) = (1+\delta')r\}$. That is, for each such point x, we choose the Voronoi ball $B_r(p)$ with p such that $d(x,p) = (1+\delta')r$ with smallest possible δ'. And we select the point x with the largest such δ'.

This landmark selection algorithm always selects landmarks on the network boundary[2] but it does not require the detection of the network boundary, nor does it require the knowledge of the medial axis and local feature size, whose computation is sensitive to noise. New landmarks in different Voronoi cells can be inserted in parallel as the algorithm executes locally inside each Voronoi cell. Thus the new landmark selection method is more robust and practical compared with the γ-sampling used in our previous algorithm. Of course when the algorithm terminates, it produces a set of landmarks L so that the Delaunay complex $DC(L)$ has both the global rigidity and the coverage property. Next, we show the algorithm terminates and has bounded landmark density.

5.2.2.4 Landmark density by incremental refinement

Here we show that every landmark q added by the incremental algorithm is not sufficiently covered by existing landmarks, i.e., the distance to its closest landmark is at least $\gamma \cdot ILFS(q)$ for an appropriate parameter $\gamma < 1/3$. If a point $x \in \partial R$ is

[2]Landmarks added by condition 1 will be on ∂R for sure. For the landmarks added by the 2nd condition, by the same argument as in Theorem 44 the points outside the Voronoi balls must be inside the pies. And the least uncovered point stays on the region boundary ∂R.

within $\gamma \cdot ILFS(x)$ from any landmark, we say x is γ-*covered*.

We first state a useful Lemma.

Lemma 46 *[Lemma 32 in Chapter 4] Given a disk B containing at least two points on ∂R , for each connected component of $B \cap R$, either it contains a point on the inner medial axis, or its intersection with ∂R is connected.*

Lemma 47 *If a Voronoi cell $V(u)$ violates the local Voronoi edge connectivity condition, the new landmark q selected is not covered by any landmark within $\gamma \cdot ILFS(q)$, for any $\gamma < 1/3$.*

Proof. Since the boundary of the Voronoi cell $V(u)$ is composed of segments on ∂R and the Voronoi edges u, $V(u)$ must have two or more connected components on the domain boundary ∂R as well. Take a Voronoi edge endpoint p that stays on a different boundary segment with u. $d(u,p) \leq d(u,q)$ since q is the furthest such endpoint. See Figure 44.

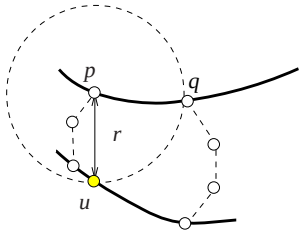


Figure 44: The new landmark q is not γ-covered for $\gamma < 1/3$.

We take a ball $B_r(p)$ with $r = d(u,p)$. $B_r(p)$ intersects the boundary ∂R in two or more connected pieces, since both u and p are inside. By Lemma 46 there is a point on the inner medial axis inside $B_r(p)$. That means $ILFS(p) < d(u,p)$. Since $IFLS$ is 1-Lipschitz, $ILFS(u) \leq IFLS(p) + d(u,p) < 2d(u,p)$. Apply this again we get $ILFS(q) \leq IFLS(u) + d(u,q) < 3d(u,q)$. Thus the claim is proved. □

Lemma 48 *If a Voronoi cell $V(u)$ for a landmark u violates the local Voronoi ball coverage condition, the new landmark q selected is not covered by any landmark within $\gamma \cdot ILFS(q)$, $\gamma = \delta/(2+\delta)$.*

Proof. If q is selected as the new landmark, $d(u,q) > (1+\delta)r$ for any Voronoi vertex p of the landmark u, $r = d(p,u)$ is the radius of the Voronoi ball at p. Now we have by triangle inequality $d(q,u) \geq d(p,q) - d(p,u) > \delta r$. That is, $r < d(q,u)/\delta$. Similar to the argument in Theorem 44, $ILFS(q) \leq d(q,u) + 2r < (1+2/\delta)d(q,u)$. Thus, $d(q,u) > \gamma \cdot ILFS(q)$ with $\gamma = \delta/(2+\delta)$. □

The above results show that our local conditions do identify points on the boundary that need to be γ-covered for $\gamma < 1/3$. If the inner local feature size for any point $x \in \partial R$ is at least ε for some fixed ε, then the incremental Delaunay refinement algorithm will eventually terminate, as every new landmark included covers at least an interval of length 2ε centered at itself. This procedure cannot go on indefinitely.

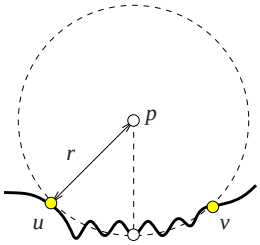


Figure 45: The landmark set may not be a γ-sample of ∂R. The local feature size for points on the segments between u, v is smaller than the distance to u or v.

We remark that the algorithm will certainly terminate when the landmark set is a γ-sample for any $\gamma < 1$, but it may also terminate before that if both the rigidity and coverage conditions are met, as shown in Figure 45. This can be understood in terms of our algorithm picking up the major geometric features and ignoring the noisy features of R. The rigidity and coverage properties guarantee that the reconstructed Delaunay complex will approximate R and are what we really care about in our localization algorithm. The γ-sample for R can be much denser than what is needed in practice.

Last we show that the landmark set generated by the incremental algorithm has bounded density.

Theorem 49 *Suppose L is the generated landmark set by the incremental algorithm. If any landmark from L is removed, then it is not a γ'-sample of ∂R, with $\gamma' = \gamma/(1+\gamma)$, $\gamma < \max(1/3, \delta/(2+\delta))$.*

Proof. For the last landmark q inserted, by Lemma 47 and Lemma 48, it is not within distance $\gamma \cdot ILFS(q)$ of any existing landmark. Since $\gamma > \gamma'$ the claim is true for q.

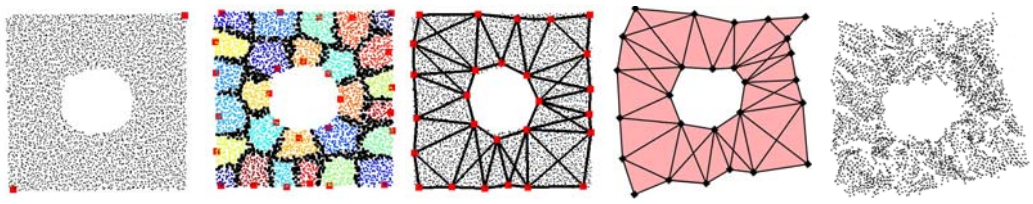
For any landmark q' added before q, we know $d(q,q') > \gamma \cdot ILFS(q)$, since q is added with q' already present in the current landmark set. Since $IFLS$ is 1-Lipschitz, we have $IFLS(q') \leq IFLS(q) + d(q,q') < (1+1/\gamma) \cdot d(q,q')$. Therefore, $d(q,q') > \gamma' \cdot IFLS(q)$. Notice that this argument is true for any pair of landmarks q, q' with q added after q'. Thus for q', the distance to any landmark in L is at least greater than $\gamma' \cdot IFLS(q')$. The claim is true. □

5.3 Incremental Delaunay refinement

5.3.1 Algorithm description

Suppose a large number of sensor nodes are scattered in a geometric region, where nearby nodes can directly communicate with each other. Similar to our previous work in Chapter 4, we do not enforce that the communication graph follows the unit disk graph model (in our simulations we use both a quasi-UDG model and a probabilistic radio model), nor do we assume any knowledge of the node locations or inter-distances. Our goal is to discover the location of the nodes in the sensor field, using the local connectivity information alone.

The basic idea is to select landmarks incrementally in the network until both the global rigidity and the coverage property are satisfied as described in Section 5.2. The biggest difference between this chapter and Chapter 4 is that the new landmark selection algorithm does not depend on the success of boundary detection or the knowledge of the local feature size. Thus the new algorithm is more robust in practice, and yet still captures the geometry of the network. Next we will explain each step of the algorithm in detail. Unless specified otherwise, all the distances are by default measured by the geodesic distance, which is approximated by



(i) (ii) (iii) (vi) (v)

Figure 46: Step by Step Incremental Delaunay Refinement Method. The number of nodes is 3887. The connectivity follows a unit disk graph model with average node degree 7.5. (i) Start with two landmarks on the boundary arbitrarily. (ii) The final Voronoi diagram when the algorithm stops. (iii) The Delaunay edges extracted from the Voronoi cells of the landmarks. (iv) Embedding Result. (v) All nodes localized.

the minimum hop count between two nodes.

5.3.1.1 Select initial landmarks

We start with two landmarks arbitrarily selected on the boundary. In order to guarantee these two starting landmarks are definitely on the boundary, we flood the network from a random node r and find the farthest node p from r, p must be on the network boundary. Then we flood from p and find the farthest node q from p. q will be on the boundary as well. We use p and q as our two initial landmarks. See Figure 46(i)

5.3.1.2 Compute Voronoi diagram

Once we have some landmarks, we calculate the landmark Voronoi diagram in a distributed way. Each landmark learns of its closest landmark(s) and all the nodes with the same closest landmark are naturally classified to be in the same Voronoi cell. Recall that the landmarks are included incrementally. A new landmark initiates a flood message which is propagated only inside its Voronoi cell—when a node receiving this message sees that its hop count to some preexisting landmark is equal or smaller, then the message is dropped. As more landmarks are included, the size of Voronoi cells decreases, and so does the communication cost.

Nodes with more than one closest landmarks lie on a Voronoi edge or vertex.

Although the straightforward definition of *Voronoi vertex* is a node with equal distance to at least three landmarks, one robustness concern is that there may not be a node that qualifies for this definition by the discrete network hop count measure. In Chapter 4, we proposed a merging heuristic to get Voronoi vertices. Here we refine this process with rigor and propose the following witness definition to guarantee the existence of Voronoi vertices.

Definition 50 *A node p is called a* 2-witness *for a pair of landmark $\{u,v\}$, if $d(p,u), d(p,v)$ are among the top m smallest hop count distances from landmarks to p and these hop count distances differ at most by β_2. β_2 is called the relaxation parameter for 2-witnesses.*

In other words, we denote by $\ell_i(p)$ the set of landmarks with the i-th smallest distance to p and $d_i(p)$ the i-th smallest distance from landmarks to p. Then a node p is the 2-witness for all pairs of landmarks in $L_2 = \cup_{i=1}^{m} \ell_i(p)$ such that $d_m(p) - d_1(p) \leq \beta_2$ and $d_{m+1}(p) - d_1(p) > \beta_2$. We call L_2 the *2-witness landmark set* for p. p witnesses every pair in L_2.

The boundary of a Voronoi cell of a landmark u is the collection of 2-witnesses with u in their landmark set. With the 2-witnesses we will detect 3-witnesses for triples of landmarks, a.k.a. the Voronoi vertices, by properly merging neighboring 2-witnesses with different landmark sets. In general we define a k-witness as follows, for $k \geq 3$.

Definition 51 *A node p is called a k-*witness *for a tuple of k landmarks, if p is a $k-1$-witness and the k landmarks are among the top m closest landmark set $L_k = \cup_{i=1}^{m} \ell_i(p)$ with $d_m(p) - d_1(p) \leq \beta_3$, $d_{m+1}(p) - d_1(p) > \beta_3$. β_k is called the relaxation parameter for k-witnesses. L_k is called the k-witness landmark set for p.*

The parameters β_k are appropriately chosen as explained below. By the analog of the continuous case, the 2-witnesses correspond to the 1-dimensional Voronoi edges. The k-witnesses for $k \geq 3$ correspond to 0-dimensional Voronoi vertices. Thus we hope that the collection of 2-witnesses for each landmark (i.e., its Voronoi edges) is connected, and that the k-witnesses with $k \geq 3$ for different k-tuples form

isolated connected components that separate 2-witness groups with different landmark set.

To show this we first give a number of observations.

Lemma 52 *If $\beta_k \geq 1$, there cannot be two neighboring k-witnesses p,q such that the landmark sets they witness do not share any common landmark.*

Proof. We will just prove this for β_2 as the proof is the same for other k. Assume by contradiction that the set of landmarks p witnesses $L_2(p)$ and the set $L_2(q)$ that q witnesses do not share any common landmark. We take $u_1 \in \ell_1(p)$ and $u_2 \in \ell_1(q)$. We have $d(q,u_2)+1 \geq d(p,u_2)$, since p,q are neighboring nodes. Also since u_2 is not among $L_2(p)$, we have $d(p,u_2) > d(p,u_1)+\beta_2$. With similar argument we have

$$\begin{array}{lll} d(q,u_2)+1 & \geq d(p,u_2) & > d(p,u_1)+\beta_2 \\ & \geq d(q,u_1)-1+\beta_2 & > d(q,u_2)+2\beta_2-1. \end{array}$$

Thus $\beta_2 < 1$. This shows a contradiction. □

Now we examine what nodes among the 2-witnesses are selected to be 3-witnesses. The Voronoi boundary of each landmark is required to be connected, therefore, we group the 2-witnesses for each landmark u by the set of landmarks they witness. Adjacent 2-witnesses that witness different landmark sets will be selected as 3-witnesses with a properly selected relaxation parameter β_3. We choose $\beta_2 = 1$.

Lemma 53 *If there are two neighboring 2-witness nodes p,q that witness different landmark set, i.e., $L_2(p) \neq L_2(q)$, and $\beta_3 = 2\beta_2+2$, p,q are both 3-witnesses of the landmarks in $L_2(p) \cup L_2(q)$.*

Proof. By Lemma 52, there is a landmark u such that $u \in L_2(p) \cap L_2(q)$. Choose $u_2 \in L_2(p) \setminus L_2(q)$ and $u_3 \in L_2(q) \setminus L_2(p)$. Now we have,

$$\begin{array}{lll} d(p,u_3) & \leq d(q,u_3)+1 & \leq d(q,u_1)+\beta_2+1 \\ & \leq d(p,u_1)+\beta_2+2 & \leq d_1(p)+2\beta_2+2. \end{array}$$

Thus $u_3 \in L_3(p)$. With a symmetric argument $u_2 \in L_3(q)$. Therefore both p and q are 3-witnesses of the landmarks in $L_2(p) \cup L_2(q)$. □

Therefore, the Voronoi boundary of a landmark will have connected components of 2-witnesses (with the same witness landmark set) connected by 3-witnesses. Intuitively, this corresponds to Voronoi edges connected by Voronoi vertices. We will perform this witness selection operation further so that among the 3-witnesses, neighboring nodes with different witness landmark sets will be identified as 4-witnesses, if $\beta_4 = 2\beta_3 + 2$. Each connected component of k-witnesses with the same landmark set will generate the corresponding Delaunay simplices. The witness identification procedure continues until the groups of k-witnesses with the same witness landmark set are isolated components.

The witness identification algorithm only uses local information. With the witnesses identified, we can output the combinatorial Delaunay complex as we will explain later.

5.3.1.3 Select more landmarks incrementally

With the Voronoi diagram from the initial 2 landmarks, we then select more landmarks incrementally. Corresponding to Section 5.2, for each landmark u and its Voronoi cell $V(u)$, we check:

- If the 2-witnesses (a.k.a. Voronoi edges) of u are not connected (this can be checked by having each connected component of the union of u's Voronoi edges send a message to u), we choose among all nodes that are endpoints of Voronoi edges lying on the network boundary[3] and select the one furthest from u as a new landmark.
- If the 2-witnesses of u are connected, we check each point p in Voronoi cell $V(u)$ and any Voronoi vertex v associated with u. We select point p as the new landmark if p is furthest away from any relaxed Voronoi ball $B_{(1+\delta)r}(v)$ among all points that are not yet δ-covered by Voronoi balls of u. Here r is the hop-count distance between u and v.

[3]Notice that we can discover such nodes as each Voronoi edge is a connected set of 2-witnesses with the same landmark set, whose endpoints are either 3-witnesses or nodes on the network boundary.

As the conditions are local, new landmarks can be selected in different Voronoi cells in parallel. The Voronoi diagram is then updated until no more landmarks are selected.

Figure 46 (ii) is the final Voronoi diagram when the landmark selection stops. The Delaunay edges extracted from the final Voronoi diagram are shown in Figure 46 (iii). When the algorithm stops, both the global rigidity and good coverage are guaranteed.

5.3.1.4 Extract Delaunay complex

When all the landmarks are in place and the final Voronoi diagram is computed, using the witnesses we identified earlier, each connected component of k-witnesses with the same landmark set will generate a corresponding Delaunay simplex. In particular, for each k-witness p, $k \geq 3$, we output for each k-tuple in the witness landmark set $L_k(p)$ a $k-1$-dimensional simplex that implicitly includes all its faces. These simplices are collected to be embedded in the next step. The embedding of the Delaunay complex is the only centralized operation in the algorithm. Once the Delaunay complex is embedded, its realization is disseminated to the entire network to localize the rest of the nodes. Notice that since the Delaunay complex is a compact structure whose size depends on the network geometric complexity, and since only Voronoi nodes are involved in embedding it, the cost of collection and dissemination is substantially smaller than the cost of collecting the entire connectivity graph for any centralized localization algorithm.

5.3.1.5 Embed Delaunay complex

In brief, we choose one simplex, embed it as a starting point, and then embed each neighboring simplex side-by-side to the one already embedded. As mentioned earlier, two k-witnesses ($k \geq 3$) with different landmark sets are connected through m-witnesses with $2 \leq m < k$. Thus each simplex we extract must share an edge with a neighboring simplex and the 2 simplices cannot overlap, so the embedding is unambiguous. For example, suppose a simplex S is already embedded, and we want to embed a neighboring simplex (triangle) S' that shares a common edge with

S. We use bilateration to find the 2 possible positions for the third landmark of S' that has not yet been embedded and choose the one that does not cause S and S' to overlap.

Since we ran our new algorithm on more complicated topologies than what our original algorithm was capable of, we encountered many high dimensional simplices (see for example the sun shape in Figure 50). In this case we embed each high-dimensional simplex using multi-lateration to the other landmarks of the simplex that are already embedded, in order to take advantage of all known distance measurements. Since we only have estimated distances, we solve the optimization problem of minimizing the mean square error among the distances as described in [83]. And as another optimization, we run a mass-spring relaxation on the simplex in order to smooth out the distance errors.

We remark that our embedding algorithm only makes sure that adjacent Delaunay triangles are embedded 'side-by-side', thereby allowing us to get a very good embedding of the network. However, it does not guarantee a planar embedding—one part of the network can still curve around and intersect with another part of the network. It is an NP-hard problem to find a planar embedding given a planar graph with specified edge lengths. A particularly challenging scenario is when embedding a network with a hole and we want to connect the loop of simplices cycling back to itself. One approach we use to prevent one simplex from landing atop another is by setting some boundary lines defined by the first embedded simplex that no other simplex may cross. If a landmark goes over this line, it is embedded to the line. This works well in many cases, and is what we used to get the result in Figure 49. If a landmark should receive more than one coordinate assignment (arising from two simplices coming around the hole), we simply embed it at the centroid of its different assignments. However the above steps do not ensure planarity, and can even introduce flipped simplices, as can be seen in the flower and music images of Figure 50, and elsewhere. We emphasize that our algorithm guarantees correct orientation of the simplices, but once other heuristics are applied, the guarantees no longer hold. It still remains for future work to develop efficient approximation algorithms with theoretical guarantees for planar graph embedding.

5.3.1.6 Network localization

When the Delaunay complex is embedded and disseminated to all the nodes, each non-landmark node uses its hop count estimation to 3 (or more) landmarks to trilaterate its own location (as in the atomic trilateration in [82]).

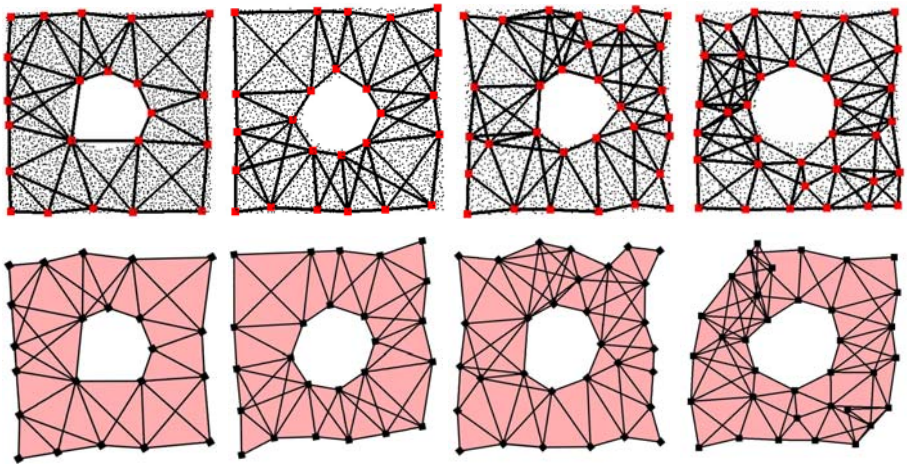
5.4 Simulation

We conducted extensive simulations under various scenarios to evaluate how well our algorithm extracts the network topology and how performance is affected by different factors such as node density, or communication model (quasi-UDG, probabilistic model, etc.). Typically our examples have an average node degree of around 10, but we also get good performance for average degree as low as 6. We also demonstrate a good result for a special case where nodes are aligned on a perfect grid having an average degree of 4. We evaluate the communication cost of our algorithm at the end.

Influence of node density. Theoretically, our algorithm performs better under higher node density since the hop-count distance between nodes is a better approximation of the geodesic distance between them.

Figure 47 shows the results of networks with different densities but with the same communication range. Notice that when the average degree is below 7, not all selected landmarks are on the boundary, as Voronoi edges may be broken at small holes in the network. The performance deteriorates when the average degree drops below 6, when error accumulation by using hop-count distance becomes too large to use for an accurate embedding.

Influence of network communication models. We also evaluate our algorithm on connectivity models other than unit disk graph model, in particular, *quasi-unit disk graph model (quasi-UDG)* and *probabilistic connectivity model*. In *quasi-UDG*, two nodes are connected by an edge if the Euclidean distance between them is no greater than a parameter α, $\alpha \leq 1$, and are not connected by an edge if the Euclidean distance is larger than 1. If the Euclidean distance d is in the range $(\alpha, 1]$, we include this edge with probability $(1-d)/(1-\alpha)$. In the *probabilistic connectivity model*,



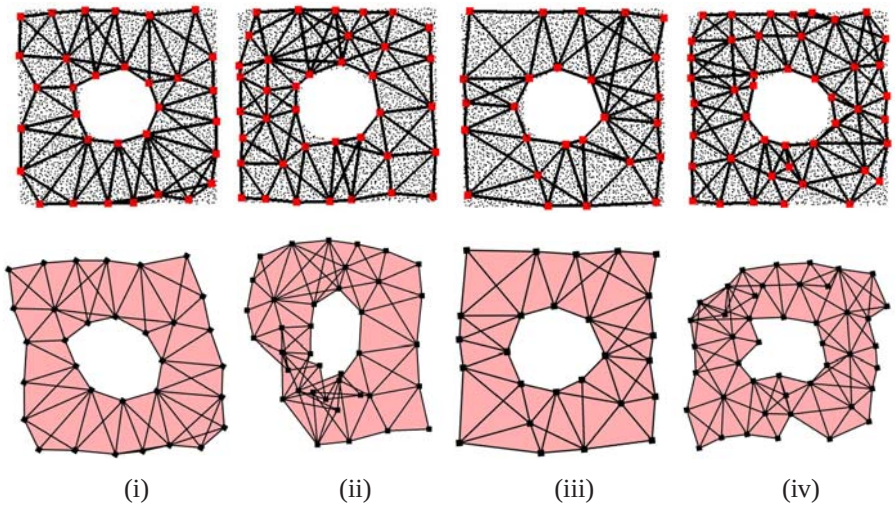
(i) (ii) (iii) (iv)

Figure 47: The embedding results for networks of different node densities. The communication ranges are the same for all 4 networks. The first row shows the ground truth; the second row shows our embedding of the landmark nodes. From left to right the models depicted have (i) 3887 nodes, avg. degree 10.28. (ii) 3044 nodes, avg. degree 7.6. (iii) 2680 nodes, avg. degree 6.3. (iv) 2320 nodes, avg. degree 5.7.

we start with the unit disk graph model and remove each edge with probability $1-\beta$.

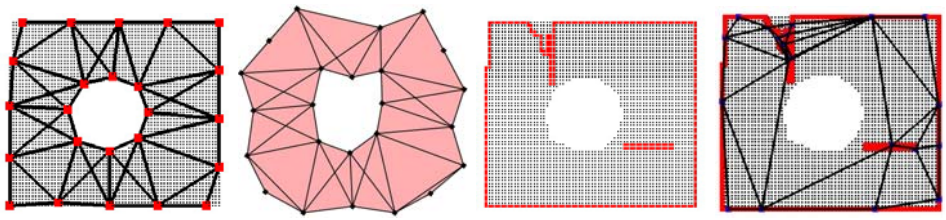
We show some representative cases in Figure 48. (i) and (ii) use the quasi-UDG model. (iii) and (iv) use the probabilistic model. We have good embedding results even when α or $\beta = 0.6$ with an average degree of around 6. When α or $\beta = 0.5$, the algorithm starts to deteriorate.

Comparison with our previous work in Chapter 4. Since the new algorithm does not depend on boundary detection, it not only avoids the computationally expensive operation of detecting the network boundary, but can work under conditions where the boundary detection would give poor results, causing an unsatisfactory outcome. Figure 49 is a network with nodes laid out on a perfect grid with an average degree of only around 4. This is an example that will not work using our previous algorithm as boundary detection will fail. As far as we know, no known boundary detection algorithm can work on networks with such low average degree. Figure 49(i)(ii) shows the ground truth and embedding result of the new algorithm. Note the low



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

Figure 48: Effect of network communication models on the embedding. The first row shows the ground truth; the second row is our embeddings of the landmark nodes. All the networks have 3887 nodes and the same communication range. From left to right the models depicted are (i) quasi-UDG model, avg. degree 6.4, $\alpha = 0.6$ (ii) quasi-UDG model, avg. degree 5.6, $\alpha = 0.5$ (iii) delete each edge with probability $1 - \beta$. $\beta = 0.6$, avg. degree 6.2 (iv) Same model as (iii), $\beta = 0.5$, avg. degree 5.0.



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

Figure 49: A perfect grid network. 3388 nodes, avg. degree 3.87. (i) the Delaunay complex extracted from the Voronoi cells of the landmarks using the new algorithm. (ii) the embedding result. (iii) the boundary detection result. (iv) the Delaunay complex result using the previous algorithm.

degree does cause some locally inaccurately embedded pieces. At two top corners, the triangles are degenerate as the hypotenuse has exactly the same length as the other 2 sides measured by hop-count in the grid network. Nevertheless we still capture the topology and the global geometry rather faithfully. Figure 49(iii) is the boundary detection result using the method in [89], which generates a Delaunay Complex that does not capture the network geometry (Figure 49(iv)).

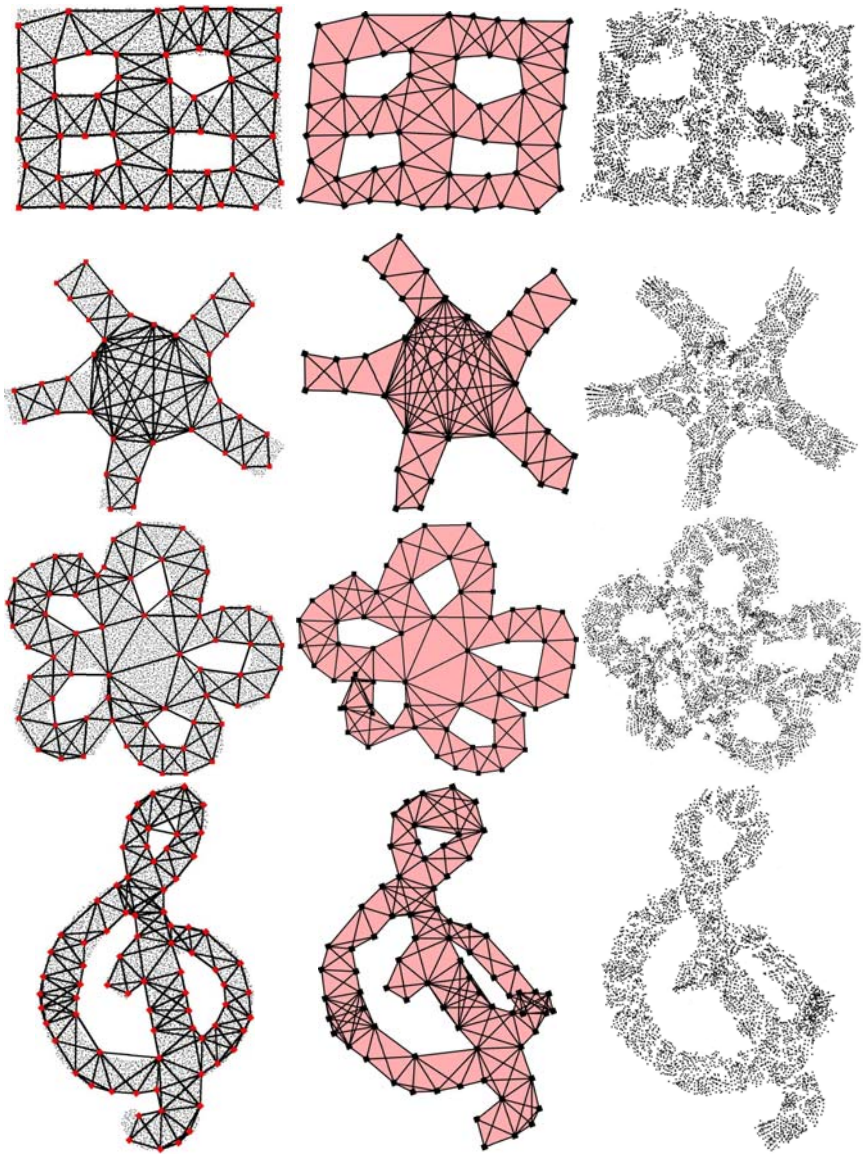
We do not explicitly compare with other localization algorithms with network connectivity information only, for example, multi-dimensional scaling (MDS) [86]. There is already a thorough comparison of MDS with our previous algorithm in Chapter 4.

Different network topology. We show more results using our algorithm for a number of networks with convoluted shapes in Figure 50.

Communication cost of the algorithm. In the execution of the incremental landmark selection algorithm, the new landmarks in different Voronoi cells are selected in parallel and each new landmark only floods locally in its Voronoi cell. In one iteration, many Voronoi cells can be refined and new landmarks selected. We use the number of nodes in each Voronoi cell to evaluate the communication cost incurred by adding each new landmark. In Figure 51, we run our algorithm on a group of networks with the same shape (similar to Figure 46), the same communication range and different node densities. We calculate the average number of nodes in each Voronoi cell in each iteration. The communication cost drops dramatically after a couple of iterations and the algorithm typically stops after a small constant number of iterations.

5.5 Conclusion

This chapter is a follow-up work of Chapter 4 [70] solving the localization problem using connectivity information only. We develop a new landmark selection algorithm using incremental Delaunay refinement method in a distributed manner. The new algorithm keeps the good properties (global rigidity and coverage) needed for localization, and yet is not dependent on network boundary detection. This allows for a more robust algorithm, less sensitive to the noisy results of boundary



(i) (ii) (iii)

Figure 50: Running our algorithm on different topologies. The first row is windows shape, 6495 nodes, avg. degree 9.97. The second row is sun shape, 5217 nodes, avg. degree 10.3. The third row is flower shape, 8350 nodes, avg. degree 9.14. The fourth row is music shape, 6176 nodes, avg. degree 10.2. Columns: (1) the ground truth. (2) the embedded landmark nodes. (3) all the nodes embedded using multi-lateration to the closest landmark nodes.

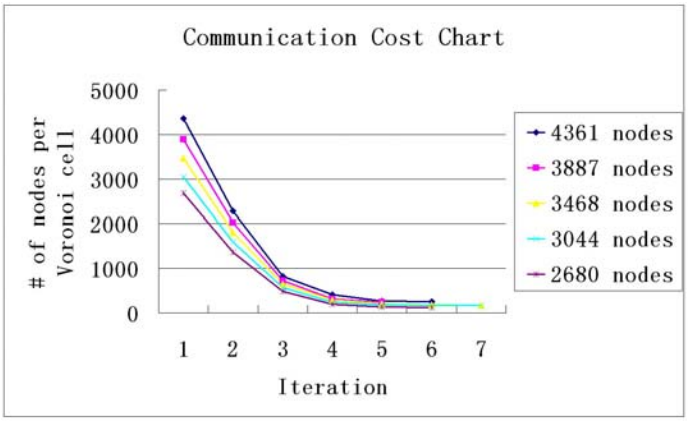


Figure 51: The average size of the Voronoi cell in each iteration until the algorithm stops. The size of the network varies from 2680 to 4361.

detection and avoids its high computation cost. Thus our new algorithm is more applicable in practice, performing well in networks with low average degree and complex shapes.

Chapter 6

Conclusions and Future Work

In this thesis, we contribute a comprehensive framework for discovery of sensor network geometry and topology, apply it to several challenging problems: Boundary Recognition, Homology Computation, Layout and Localization. Specifically, this thesis involves the following works: 1). to develop a simple, distributed algorithm to detect the nodes on the boundaries in a sensor network by using only connectivity information. We can connect those boundary nodes into meaningful boundary cycles and obtain the medial axis of the sensor field as a byproduct. 2). to introduce a new feature size for bounded domains, which measures the size of the smallest topological feature of the domain. From this feature size, combined with the concepts of geodesic Delaunay triangulations and witness complexes, we can compute a homology of the original domain. 3). to propose a distributed algorithm to discover and recover the layout of a large sensor network having a complex shape, avoid global flips where a part of the network folds on top of another. Based on the layout, we will also get a practical and accurate localization algorithm for large networks, with only network connectivity information. 4). to develop a new landmark selection algorithm with incremental Delaunay refinement as a follow up to the previous algorithm. The new algorithm does not assume any knowledge of the network boundary and runs in a distributed manner to select landmarks incrementally until both the global rigidity property and the coverage property are met. All of our works have the following properties: Only connectivity information is required. They do not depend on Unit Disk Graph model. All methods will work

even for networks with low degree and low density. These properties reflect the requirements from practical scenario.

There are lots of potential working directions as follow-up of our works in the field of geometry and topology discovery based only on connectivity information.

1. Our current works only consider static scenario, one direction of future works is to extend them to mobile sensor networks such as applying current boundary detection algorithm to mobile scenarios, then we can make mobile nodes map an unknown environment.
2. All our current works are limited in planar graph. A natural idea is to investigate the possible solutions for these problems in 2-Dimension manifold or 3-Dimension. Some ideas in this thesis also can be applied on Surface Reconstruction problem in Graphics area.
3. There are more theoretical works to be done. For both our boundary detection work as well as layout and localization work, we proved rigorously the correctness of the algorithms for a continuous geometric domain case. One of the future works is to give a theoretical performance guarantee for the discrete network under mild assumptions of node density and communication models.
4. How to improve the performance of our results for very sparse network? Our methods will work very well for the network with average degree around 6 or higher, which is the most preferred average degree in practical network. However, there are still a lot of real scenarios which could have lower average degree in sparser network. No known efficient ways to solve geometry and topology discovery problems in very sparse networks yet.

Bibliography

[1] S. Alstrup, C. Gavoille, H. Kaplan, and T. Rauhe. Nearest common ancestors: a survey and a new distributed algorithm. In *SPAA '02: Proceedings of the 14th annual ACM symposium on Parallel algorithms and architectures*, pages 258–264, 2002.

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

[3] 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.

[4] N. Amenta and R. K. Kolluri. Accurate and efficient unions of balls. In *SCG '00: Proceedings of the 16th annual Symposium on Computational Geometry*, pages 119–128, 2000.

[5] B. D. O. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang. Graphical properties of easily localizable sensor networks. *Wireless Networks*, pages 177–191, 2007.

[6] J. Aspnes, D. Goldenberg, and Y. R. Yang. On the computational complexity of sensor network localization. In *ALGOSENSORS '04: The 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pages 32–44, 2004.

[7] D. Attali, H. Edelsbrunner, and Y. Mileyko. Weak witnesses for Delaunay triangulations of submanifolds. In *SPM '07: Proceedings of ACM Symposium on Solid and Physical Modeling*, pages 143–150, 2007.

[8] M. Badoiu, E. D. Demaine, M. T. Hajiaghayi, and P. Indyk. Low-dimensional embedding with extra information. In *SCG '04: Proceedings of the 20th annual Symposium on Computational Geometry*, pages 320–329, 2004.

[9] M. Bateni, E. D. Demaine, M. Hajiaghayi, and M. Moharrami. Plane embeddings of planar graph metrics. *Discrete Computational Geometry*, 38(3):615–637, 2007.

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

[11] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 46–54, 2004.

[12] A. Björner. Topological methods. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, chapter 34, pages 1819–1872. North-Holland, Amsterdam, 1995.

[13] A. I. Bobenko and B. A. Springborn. A discrete laplace-beltrami operator for simplicial surfaces. Technical Report, 2005.

[14] J.-D. Boissonnat, L. J. Guibas, and S. Y. Oudot. Manifold reconstruction in arbitrary dimensions using witness complexes. In *SCG '07: Proceedings of 23rd annual Symposium on Computational Geometry*, pages 194–203, 2007.

[15] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of Lipschitz surfaces. In *SCG '06: Proceedings of the 22th annual Symposium on Computational Geometry*, pages 337–346, 2006.

[16] K. Borsuk. On the imbeddings of systems of compacta in simplicial complexes. *Fundamenta Mathematicae*, 35:217–234, 1948.

[17] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

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

[19] J. Bruck, J. Gao, and A. Jiang. MAP: Medial axis based geometric routing in sensor networks. In *MOBICOM '05: Proceedings of the 11th annual International Conference on Mobile Computing and Networking*, pages 88–102, 2005.

[20] D. Burago, Y. Burago, and S. Ivanov. *A Course in Metric Geometry*, volume 33 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2001.

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

[22] G. Carlsson and V. de Silva. Topological approximation by small simplicial complexes. In *PBG '04: Proceedings of the Symposium on Point-Based Graphics*, June 2004.

[23] F. Chazal, D. Cohen-Steiner, L. J. Guibas, and S. Y. Oudot. The stability of persistence diagrams revisited. Research Report 6568, INRIA, July 2008.

[24] F. Chazal, D. Cohen-Steiner, and A. Lieutier. A sampling theory for compact sets in Euclidean space. In *SCG '06: Proceedings of the 22nd annual Symposium on Computational Geometry*, pages 319–326, 2006.

[25] F. Chazal and A. Lieutier. Weak feature size and persistent homology: Computing homology of solids in $\mathbb{R}^n$ from noisy data samples. In *SCG '05: Proceedings of the 21st annual Symposium on Computational Geometry*, pages 255–262, 2005.

[26] S.-W. Cheng, T. K. Dey, and E. A. Ramos. Manifold reconstruction from point samples. In *SODA '05: Proceedings of 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1027, 2005.

[27] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *SCG '05: Proceedings of the 21st annual Symposium on Computational Geometry*, pages 263–271, 2005.

[28] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.

[29] V. de Silva. A weak definition of Delaunay triangulation. Manuscript. Preprint available at `http://math.stanford.edu/comptop/preprints/weak.pdf`, October 2003.

[30] V. de Silva and G. Carlsson. Topological estimation using witness complexes. In *PBG '04: Proceedings of the Symposium on Point-Based Graphics*, pages 157–166, 2004.

[31] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry: Theory and Applications*, 35(1-2):124–141, 2006.

[32] R. Dyer, H. Zhang, and T. Moeller. Delaunay mesh construction. In *SGP '07: Proceedings of Eurographics Symposium on Geometry Processing*, pages 273–282, 2007.

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

[34] J. Elson. *Time Synchronization in Wireless Sensor Networks*. PhD thesis, University of California, Los Angeles, May 2003.

[35] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, S. Morse, B. Anderson, and P. Belhumeur. Rigidity, computation, and randomization of network localization. In *INFOCOM '04: Proceedings of the 23th Conference on Computer Communications*, pages 2673–2684, 2004.

[36] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. In *Mobile Networks and Applications*, volume 11, pages 187–200, 2006.

[37] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang. GLIDER: Gradient landmark-based distributed routing for sensor networks. In *INFOCOM '05: Proceedings of the 24th Conference on Computer Communications*, volume 1, pages 339–350, 2005.

[38] Q. Fang, J. Gao, and L. J. Guibas. Landmark-based information storage and retrieval in sensor networks. In *INFOCOM '06: Proceedings of the 25th Conference on Computer Communications*, 2006.

[39] S. P. Fekete, M. Kaufmann, A. Kröller, and N. Lehmann. A new approach for boundary recognition in geometric sensor networks. In *CCCG '05: Proceedings 17th Canadian Conference on Computational Geometry*, pages 82–85, 2005.

[40] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In *ALGOSENSORS '04: The 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pages 123–136, 2004.

[41] M. Fisher, B. Springborn, A. I. Bobenko, and P. Schröder. An algorithm for the construction of intrinsic delaunay triangulations with applications to digital geometry processing. In *SIGGRAPH Courses*, pages 69–74, 2006.

[42] D. Freedman and C. Chen. Measuring and localizing homology classes. Technical Report, Rensselaer Polytechnic Institute, May 2007.

[43] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[44] 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.

[45] S. Funke and C. Klein. Hole detection or: "how much geometry hides in connectivity?". In *SCG '06: Proceedings of the 22nd annual Symposium on Computational Geometry*, pages 377–385, 2006.

[46] S. Funke and N. Milosavljević. Guaranteed-delivery geographic routing under uncertain node locations. In *INFOCOM '07: Proceedings of the 26th Conference on Computer Communications*, pages 1244–1252, 2007.

[47] S. Funke and N. Milosavljevic. Network sketching or: "how much geometry hides in connectivity? - part II". In *SODA '07: Proceedings of 18th ACM-SIAM Symposium on Discrete Algorithms*, pages 958–967, 2007.

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

[49] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical Report UCLA/CSD-TR 02-0013, UCLA, 2002.

[50] J. Gao, L. Guibas, S. Oudot, and Y. Wang. Geodesic delaunay triangulation and witness complex in the plane. In *SODA '08: Proceedings of 19th ACM-SIAM Symposium on Discrete Algorithms*, pages 571–580, 2008.

[51] J. Gao, L. Guibas, S. Oudot, and Y. Wang. Geodesic delaunay triangulations in bounded planar domains. In *special issue of Transactions on Algorithms on SODA'08*, to appear.

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

[53] 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 *MOBICOM '06: Proceedings of the 12th annual International Conference on Mobile Computing and Networking*, pages 110–121, 2006.

[54] 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 *INFOCOM '05: Proceedings of the 24th Conference on Computer Communications*, pages 313–326, 2005.

[55] C. Gotsman and Y. Koren. Distributed graph layout for sensor networks. In *GD '04: Proceedings of the International Symposium on Grpah Drawing*, pages 273–284, September 2004.

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

[57] L. G. Guibas and S. Y. Oudot. Reconstruction using witness complexes. In *SODA '07: Proceedings of 18th ACM-SIAM Symposium on Discrete Algorithms*, pages 1076–1085, 2007.

[58] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

[59] M. Held. Voronoi diagrams and offset curves of curvilinear polygons. *Computer Aided Design*, 30(4):287–300, Apr. 1998.

[60] B. Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84, 1992.

[61] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.

[62] A. Howard, M. Mataric, and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *IROS '01: In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1055–1060, 2001.

[63] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[64] B. Karp and H. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *MOBICOM '00: Proceedings of the 6th annual International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[65] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *NSDI '05: Proceedings of the 2nd USENIX/ACM Symposium on Networked System Design and Implementation*, pages 217–230, May 2005.

[66] S. G. Kobourov, A. Efrat, D. Forrester, and A. Iyer. Force-directed approaches to sensor network localization. In *ALENEX '06: 8th Workshop on Algorithm Engineering and Experiments*, pages 108–118, 2006.

[67] A. Kolmogorov and V. Tikhomirov. ε-entropy and ε-capacity of sets of functions. *Translations of the AMS*, 17:277–364, 1961.

[68] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *SODA '06: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1000–1009, 2006.

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

[70] S. Lederer, Y. Wang, and J. Gao. Connectivity-based localization of large scale sensor networks with complex shape. In *INFOCOM '08: Proceedings of the 27th Conference on Computer Communications*, pages 789–797, 2008.

[71] G. Leibon and D. Letscher. Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In *SCG '00: Proceedings of the 16th annual Symposium on Computational Geometry*, pages 341–349, 2000.

[72] J. S. B. Mitchell. A new algorithm for shortest paths among obstacles in the plane. *Annals of Mathematics and Artificial Intelligence*, 3:83–106, 1991.

[73] J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[74] 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.

[75] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *INFOCOM '03: Proceedings of the 22th Conference on Computer Communications*, volume 22(1), pages 1734–1743, 2003.

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

[77] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete Computational Geometry*, to appear.

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

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

[80] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *MOBICOM '03: Proceedings of the 9th annual International Conference on Mobile Computing and Networking*, pages 96–108, 2003.

[81] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *ICC '01: Proceedings of IEEE International Conference on Communications*, pages 857–861, 2001.

[82] 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 annual International Conference on Mobile Computing and Networking*, pages 166–179, 2001.

[83] A. Savvides, H. Park, and M. B. Srivastava. The n-hop multilateration primitive for node localization problems. *Mobile Networks and Applications*, 8(4):443–451, 2003.

[84] A. Savvides and M. B. Strivastava. Distributed fine-grained localization in ad-hoc networks. *IEEE Transactions of Mobile Computing*, 2003.

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

[86] 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 and Computing*, pages 201–212, 2003.

[87] A. M.-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. In *SODA '05: Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms*, pages 405–414, 2005.

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

[89] Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *MOBICOM '06: Proceedings of the 12th annual International Conference on Mobile Computing and Networking*, pages 122–133, 2006.

[90] Y. Wang, S. Lederer, and J. Gao. Connectivity-based sensor network localization with incremental delaunay refinement method. In *INFOCOM '09: Proceedings of the 28th Annual IEEE Conference on Computer Communications*, 2009.

[91] K. Whitehouse and D. Culler. A robustness analysis of multi-hop ranging-based localization approximations. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pages 317–325, 2006.

[92] W.-T. Wu. On a theorem of Leray. *Chinese Mathematics*, 2:398–410, 1962.

[93] X. Zhu, R. Sarkar, and J. Gao. Shape segmentation and applications in sensor networks. In *INFOCOM '07: Proceedings of the 26th Conference on Computer Communications*, pages 1838–1846, 2007.

[94] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete Computational Geometry*, 33(2):249–274, 2005.