Network Sketching or:
"How Much Geometry Hides in Connectivity? – Part II"

Stefan Funke*   Nikola Milosavljevic†

Abstract

Wireless sensor networks typically consist of small, very simple network nodes without any positioning device like GPS. After an initialization phase, the nodes know with whom they can talk directly, but have no idea about their relative geographic locations. We examine how much geometry information is nevertheless hidden in the communication graph of the network: Assuming that the connectivity is determined by the well-known unit-disk graph model, we show that using a very simple distributed algorithm we can identify a large, provably planar subgraph of the communication graph that faithfully reflects the topology of the network. This planar subgraph can then be embedded using a simple distributed rubber-banding procedure, finally obtaining *virtual coordinates* for the nodes of the subgraph which can be instrumented for various protocols based on geographic location information. That is, there is enough geometry information hidden in the connectivity structure not only to identify topological features like network holes (as it was also exhibited in the predecessor paper [7]) but even enough information to compute a *sketch* of the network layout. Our simulation results indicate that the algorithm works very well even for very sparse network deployments and produces network sketches that come close to the original layout.

1 Introduction

Imagine the following scenario: during a long summer drought, forest fires have started in a large region of a remote nature preserve that is hardly accessible by ground transportation. To be able to continuously assess the situation and plan appropriate countermeasures, airplanes are sent out to deploy thousands of wireless sensor nodes. Due to cost restrictions and to achieve the maximum life-time by energy savings, these sensor nodes are rather low-capability devices equipped just with temperature and humidity sensors, a simple processing unit, and a small radio device that allows for communication between nearby sensor nodes. One of the first goals is now to have this network organize itself such that messages are routed within the network, regions of interest (e.g. the current fire-front) can be identified, and gathered data can be efficiently queried.

Achieving this goal becomes quite challenging since the only information a node has about the global network topology are its immediate neighbors with whom it can communicate. Lacking an energy-hungry GPS unit and being deployed from an airplane in a rather uncontrolled fashion, none of the sensor nodes is aware of its geographic location.

Assume the area of interest is some region $\mathcal{R}$. The airplanes have deployed sufficiently many sensors such that – if all of the sensors were in operation after reaching the ground – the area of interest is completely monitored by the sensors. Formally we have that for every point $p \in \mathcal{R}$ there would be at least one sensor s within Euclidean distance $|ps| \leq r_{\text{sense}}$. Where r_{sense} is the *sensing radius* of the sensor nodes, i.e. the radius within which they can monitor or estimate temperature or humidity. Unfortunately, not all sensors will be operational upon reaching the ground. Some of them might fall right into the flames and be destroyed, others might plunge into a lake or pond and be unable to perform their monitoring task. Paradoxically, we are particularly interested in those areas where there's an ongoing fire (and maybe also where there is a lake or pond), but sensor nodes that fell into these areas are unable to report this fact.

We are interested in detecting the presence and relative locations of such holes in the monitored space created by fire or other phenomena and aim for a a rough sketch of how the network nodes are laid out in the area of interest. The only information that is

*Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany; funke@mpi-inf.mpg.de. Part of this work was done while the first author was member of the Guibas Lab at Stanford University. The first author was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by the German Federal Ministry of Education and Research (FKZ 01IMC01).

†Computer Science Department, Stanford University, Stanford, CA, U.S.A; nikolam@cs.stanford.edu. The second author was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by the German Federal Ministry of Education and Research (FKZ 01IMC01).

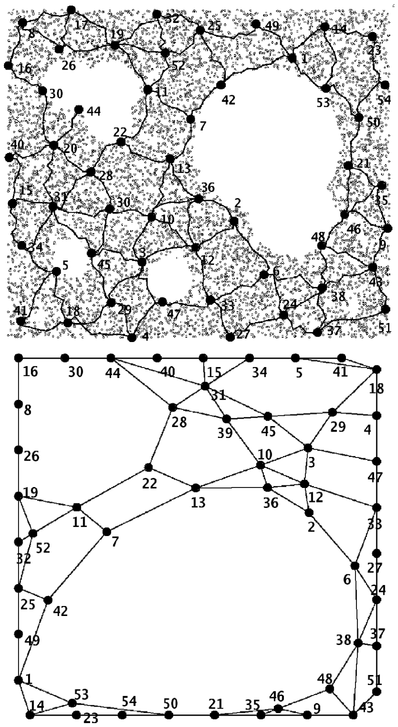

Figure 1: Based just on the *connectivity* of the network our algorithm exhibits a provably planar subgraph within the network and computes an embedding thereof. The numbers denote corresponding nodes in the embedding and the original node distribution (which is unknown to our algorithm). Notice that although the two layouts are different in terms of absolute node coordinates (due to the degrees of freedom involved with embedding the perimeter nodes on the square boundary), the network sketch is topologically correct, and the *relative* node positions are well preserved. In this case, actually, the node positions differ (roughly) by a global 90-degree rotation and a horizontal flip. Observe that node 20 and 44 are mapped on the same position since node 44 is only connected to node 20.

available to us for this task is the the *communication graph* of the wireless nodes. The *communication graph* of a wireless network has a node for each wireless station and an (unweighted) edge between two nodes if the respective stations can communicate with each other. For simplicity let us assume that two nodes can communicate with each other if they are within distance of at most 1 (communication radius). So the communication graph is a *unit disk graph* (UDG).

Related Work Recently there have been a number of papers that aim at extracting structural information of the network by analyzing its communication graph, in particular the detection of *holes* has been a target of several papers. The predecessor of this paper ([7]) aims solely at identifying nodes close to the boundary of network holes, no embedding of the identified nodes is computed, and no relative location information amongst the wholes can be extracted. The proposed algorithm is based on detecting the breakage of wave propagation contours. Correctness of the algorithm can proved under very pessimistic assumptions on the node density and hole distribution, but even in practice the algorithm requires a relatively high average node degree of the communication graph of around 20. Fekete et al. in [6] describe a method to identify boundaries in a wireless network which also does *not* require the nodes to be aware of their geographic position; their method assumes a *uniform* distribution of the network nodes in non-hole areas, though. In a more recent paper [5] the same authors present a deterministic approach for boundary recognition which does not rely on a uniform node distribution. Their paper also proposes interesting methods to aggregate the information gathered by the boundary recognition step in a higher-level topology sketch of the network. The topology sketch computed by their algorithm is a "raw" graph without an embedding, though. Their algorithm seems to require a similarly high average node degree as the approach in [7]. Very recently, an interesting new algorithm for detecting holes has been presented in [18] which is based on analyzing the homotopy types of shortest paths within the network. It appears to require the lowest node density of all the approaches presented so far. We want to emphasize that for all approaches [7, 6, 5, 18] no real geometry information like relative positions of (some of) the network nodes is computed. In some sense, the images of the computed topology sketches or determined hole boundaries included in these papers are somewhat misleading as they are drawn with the knowledge of the real node locations. The information computed by these algorithms does not suffice to produce such drawings.

Our method is based on *embedding* an appropriately chosen graph in a geometric space – specifically, a planar subgraph of the communication graph into the Euclidean plane. The graph embedding techniques have been frequently used in wireless network research for bridging the gap between geometry and connectivity. This is often labeled as computing the *virtual coordinates* of the nodes, instead of the real ones.

The method of Moscibroda *et al.* [12] attempts to compute a *low-distortion embedding* of the whole network into a single virtual space, using the theory of metric embeddings and convex optimization. The low-distortion requirement typically refers to the edge lengths, the main purpose being the preservation of path lengths. Indeed, this property is crucial for some applications (e.g. geographic point-to-point routing), where the loss incurred by the absence of location information directly relates to the embedding distortion. Unfortunately, this requires placing the graph in a very high-dimensional space (dimension is roughly logarithmic in the network size). As a consequence, certain well-established algorithms for wireless networks (e.g. face routing) do not work on such graphs, since they are designed for deployments in the plane. Moreover, the embedding algorithms are expensive in terms of time and energy, since they require non-local communication over many rounds of optimization. Finally, the best known distortion for unit-disk graphs [12] is far from practical – it is given by a multiplicative factor which depends polylogarithmically on the network size. The situation is similar for planar graphs – the best known result (distortion $O(\log^{1/2} n)$ where n is the number of nodes) is due to Rao [16] and is known to be the best possible [13], even for series-parallel graphs and even if the dimension of the Euclidean space in not restricted. Since we are only interested in two-dimensional embeddings, we drop the distance preservation requirement and instead seek to preserve topology.

Techniques somewhat similar to ours are used in ISOMAP [17] for *non-linear dimensionality reduction*, that is embedding a high-dimensional point set into a low-dimensional one while preserving as much as possible the *intrinsic* distances between the points – the lengths of the shortest paths within a low-dimensional manifold which is assumed to contain the input points. ISOMAP builds a neighborhood-based graph on the high-dimensional points, and applies a classical *linear* dimensionality reduction technique (multidimensional scaling [11]) to compute the best (in the least-squares sense) approximation of its shortest path metric. Since we are more interested in preserving topology than geometry, and do not have any (not even high-dimensional) coordinates as part of input, the graph that we build and the method that we use for embedding are different.

Our planar graph extraction technique (Section 2.1) is similar to witness complexes of de Silva and Carlsson [2], in which $k \geq 2$ points in the input dataset define a $(k-1)$-simplex if they are the k nearest neighbors of some other datapoint. However, our construction is based on shortest path distances in an associated graph rather than Euclidean distances, since the latter is not a part of our algorithm's input. Recent work of de Silva and Ghrist [3] addresses similar problems of topology discovery in location-free networks using methods of algebraic topology, more precisely the homology of certain simplicial complexes derived from the communication graph. Because the approach is not based on planar graphs, but on more complicated higher-dimensional complexes, it does not seem to allow for an efficient embedding procedure as a way of recovering the originally unknown geometry of the deployment.

Most related to our approach of computing an embedding of a communication graph is the work by Rao *et al.* [15] who propose to compute a two-dimensional embedding of the entire communication graph in a distributed way. Their algorithm has no upper bound on the distortion, but the authors show experimental evidence that it resembles the original layout reasonably well. Our method uses the same embedding on a different graph.

Our Contribution In this paper we present a very simple algorithm for sketching the relative locations of nodes and topological features in a wireless network that is represented purely by its communication graph. If the structure of the communication graph is a (quasi-)unit-disk graph determined by the geographic locations of the nodes, our algorithm extracts a provably planar graph whose face structure when embedded at the original positions faithfully represents the topology of the network. Since our planar graph even in practice is far from being a tree-like structure but rather dense, the embedding of this graph via simple methods like Tutte's algorithm yields a reasonably good sketch of the relative positions of nodes and topological features like holes in the network. As a side effect, the fact that we embed a planar graph allows the immediate application of the constructed virtual coordinates in greedy geographic routing schemes with guaranteed delivery. Our simulation results show that the algorithm works very well even in networks of very low density.

2 Planar Graph Extraction and Embedding

The core of our approach is to first compute a partition of the network into small *tiles* by choosing a set of landmarks and assigning each node to its closest (in hop-distance) landmark – like [14] we call the resulting decomposition of the network *landmark Voronoi complex (LVC)*. Its respective dual is called the *combinatorial Delaunay graph (CDG)* and captures the adjacencies of the tiles: two tiles τ_1, τ_2 are called adjacent if there exist two nodes $p \in \tau_1$ and $q \in \tau_2$ and (p, q) is an edge in the communication graph. Unfortunately, even in practical instances the CDG is not a planar graph, see Figure 2.

The core and main contribution of this paper is a way of extracting a planar subgraph from the CDG – we call it *combinatorial Delaunay map (CDM)* as well as computing a planar geometric embedding of the latter[1].

[1] A planar geometric embedding is an assignment of coordi-

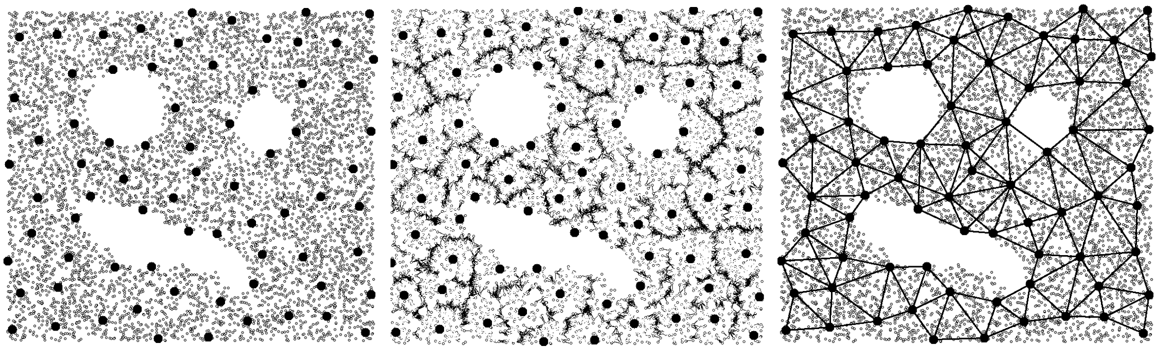

Figure 2: A network topology with a set of landmarks, the induced graph Voronoi diagram, and its dual combinatorial Delaunay graph (which is not planar!).

2.1 Planar Graph Extraction The idea for construction and the main properties of our planar graph construction are largely derived from the geometric intuition. To be specific, the planarity follows from the fact that our CDM is the *dual graph* of a suitably defined partition of the plane into simply connected disjoint regions. In the following we define such a planar partition based on a landmark set, and propose a method for identifying a subset of edges of the combinatorial Delaunay graph using only the information available in the graph connectivity. The whole reasoning is based on the fact that the original communication graph is not an arbitrary graph but in some way resembles the geometry of the underlying domain by being either a unit-disk or quasi-unit-disk graph.

In a first step we choose a set of landmarks from the set of nodes in the communication graph. To this end, we fix a small constant k, like $k = 5$, and select the set of landmarks to be a maximal k-hop independent set I_k in the communication graph, that is, for all $p, q \in I_k$, the hop-distance in the graph is more than k and $\forall s \in V - I_k$:, $I_k \cup \{s\}$ is not a k-hop independent set. This can easily be computed in linear time via a greedy algorithm, which iteratively selects a node into I_k and removes its k-hop neighborhood from the graph. A very simple distributed implementation is also rather straightforward.

We then compute the *graph Voronoi diagram* [4] of the communication graph with respect to the set of landmarks selected as above. Graph Voronoi diagram is a graph analogue of the corresponding object defined in geometry. Informally, the Voronoi diagram of a graph with respect to a subset of its vertices (the landmarks) is a partition of the vertex set into disjoint subsets (Voronoi regions) according to which landmark is closest to the vertices of a given subset. The distances are defined in a standard graph-theoretic sense, as shortest path lengths. Note that Voronoi regions and landmarks are in one to one correspondence, as in the geometric case. The Voronoi regions determined in this fashion are now the tiles.

First we introduce a *labeling of the communication graph* for a given set of landmarks.

DEFINITION 2.1.

(i) Consider a landmark a and a vertex v. We say that v is an a-vertex if a is one of the landmarks which are closest to v, and it has the smallest ID among all such landmarks.

(ii) Consider arbitrary landmarks a, b and an edge $e = (u, v)$. We say that e is an a-edge if both u and v are a-vertices. We say that e is an ab-edge if u is an a-vertex and v is a b-vertex or vice versa.

Clearly, this rule assigns unique label to each vertex and edge, due to the uniqueness of nodes' IDs. Also note that any landmark a is an a-vertex.

Next we present a criterion for making two landmarks adjacent in the CDM that we construct as a subgraph of the CDG.

DEFINITION 2.2. *Landmarks a and b are adjacent in the CDM if there exists a path from a to b whose 1-hop neighborhood (including the path itself) consists only of a and b vertices, and such that in the ordering of the nodes on the path (starting with a and ending with b) all a-nodes precede all b-nodes.*

Note that the CDG is actually defined in a very similar way with the only difference that the 1-hop neighborhood of the path is not cared about.

nates to the node of the graph such that edges in the graph do not intersect when drawn as straight line segments.

Now we prove that the landmark graph defined as above is indeed planar, for any unit-disk or quasi-unit-disk[2] (with $\alpha \leq \sqrt{2}$) embedding. The basic idea is to show that the adjacencies of Definition 2.2 are in fact adjacencies between planar regions in a suitably defined planar partition. Then planarity follows from the well known property that the dual graph of a planar partition is planar.

To define the partition, we use a *labeling of an embedding*, which of course can be only conceptually defined, and not explicitly computed, since the geometric realization is unknown and in general not unique, given only the connectivity information. However, for us it is enough that such a construction exists, because we will use only properties that hold regardless of the actual embedding.

Definition 2.3. *Consider a point x in the plane, which lies on the geometric realization of an edge e in a unit-disk realization of the communication graph. We say that e labels x as an a-point if the endpoint of e closest to x is an a-vertex. By convention, the midpoint is closest to the endpoint that has higher ID.*

Notice that in this way the same point may be labeled differently by several edges that cross in that point. Here is an important property of this labeling scheme.

Lemma 2.1. *If two edges e_1 and e_2 cross in the geometric realization, and label a shared point x as an a-point and b-point, respectively, then e_1 and e_2 are both adjacent to the same ab-edge for any quasi-unit-disk graph with $\alpha \leq \sqrt{2}$.*

Proof. Let y and z be the endpoints of e_1 and e_2, respectively, which are closer to x. The point-labeling rule (Definition 2.3) implies that y is an a-vertex and z is a b-vertex. If y and z were not adjacent, then the two opposite endpoints would not be adjacent either (because their distance is at least as big). This configuration cannot happen in either unit-disk graphs, or quasi-unit-disk graphs with parameter $\alpha \leq \sqrt{2}$ (this directly follows from the Theorem 3.5 of [1]), so this Lemma and our results theoretically hold for exactly these classes of graphs. Therefore, (y, z) is an ab-edge adjacent to both e_1 and e_2. Note that the proof is valid even in the degenerate cases: if the edges intersect along a line segment, or at a point that represents a vertex of one or both edges. ■

[2]The *quasi-unit-disk* graph ([10]) with parameter α for a set of points in the plane certainly has an edge between all nodes that have distance at most $1/\alpha$ and certainly no edge between pairs of nodes at distance more than 1. For nodes at distance between $1/\alpha$ and 1 the the presence of an edge is uncertain.

Now let us look at the implications of the adjacency criterion (Definition 2.2) in this geometric labeling.

Lemma 2.2. *If landmarks a and b are adjacent according to Definition 2.2, then there exists a labeling of points in the plane and a path in the plane from (the geometric realization of) a to (the geometric realization of) b, which is a concatenation of a subpath of a-points and a subpath of b-points.*

Proof. Consider the path from a to b referred to in Definition 2.2 and the labels of the point in the plane assigned by the edges of this path.

First modify this labeling to eliminate conflicting assignments due to the self-intersections of the path. Start following the path, say from a, and whenever an intersection is encountered, un-label the points corresponding to the section of the path between the intersecting points. For the intersecting point itself, remove the label corresponding to the landmark with smaller ID. Obviously, the points now have unique labels from the point of view of the path from a to b.

Also, it can be easily shown (by induction on the number of un-labeling operations) that this operation preserves the pattern of point labels, i.e. the resulting path from the embedding of a to the embedding of b is always a subpath of a-points concatenated with a subpath of b-points.

It may still be that a single point in the plane is assigned multiple labels by multiple paths, that is the paths that correspond to two or more distinct adjacencies in the landmark graph. Next we show that this cannot happen, i.e. when the labels that correspond to different adjacencies are overlaid in the plane, the number of different labels received by any point is at most one. Suppose without loss of generality that an a-point on the path from a to b ($a \neq b$) is the same as a c-point on a path from c to d ($\{a, b\} \neq \{c, d\}$). Consider the two edges in the *original* paths (before the un-labeling) that this point lies on (if there are more than two, choose arbitrarily one edge from each path). By Lemma 2.1 there is an ac-edge in the 1-hop neighborhood of both paths. It must be $c = b$, by construction of the path from a to b. Also, it must be $d = a$, by construction of the path from c to d. But this violates the initial assumption $\{a, b\} \neq \{c, d\}$, and the proof is complete. ■

Now we can state the main planarity result.

Theorem 2.1. *The combinatorial Delaunay map (CDM) built using the rule of Definition 2.2 is planar for any quasi-unit disk graph with $\alpha \leq \sqrt{2}$.*

Proof. We show that the landmark graph is in fact a subgraph of the dual graph corresponding to a partition of the plane. Lemma 2.2 implies that the points with the same label form a connected region (any two points of this region have a path to the embedding of the corresponding landmark). The regions that correspond to different labels are disjoint, since every point gets at most one label. Consider a partition of the plane into the labeled regions described above, and the remaining un-labeled regions. Obviously, every adjacency in the landmark graph can be matched to one adjacency between the regions and vice versa. Hence the landmark graph is planar, as it is a subgraph of the dual graph of a planar partition. ∎

2.2 Properties of the CDM To illustrate why we believe that the CDM faithfully reflects the topology of the network let us consider the setting where infinitely many network nodes are distributed over some region in the plane at infinite density. A finite set of those nodes is then extracted by selecting nodes greedily and always removing all other nodes within distance k. The property of the resulting set L of nodes is that for any point p in the plane there is some $l \in L$ with $d(l,p) \le k$ (covering property) and, of course $\forall l_1, l_2 \in L$ we have $d(l_1, l_2) > k$ (packing property). Now consider the *Voronoi diagram* of the set L of points in the plane and a subgraph of the respective dual graph which is defined as follows: two nodes $l_1, l_2 \in L$ are adjacent in the *conservative Delaunay graph* $\mathcal{CD}$ if the Voronoi edge between l_1 and l_2 has length at least $3 + \epsilon$. We now want to argue that $\mathcal{CD}$ is well connected, in the sense that each face of the the plane decomposition that it determines, although not necessarily a triangle, is bounded by only a few edges.

LEMMA 2.3. *If $k \ge 290$ and sufficiently small ϵ the conservative Delaunay graph as defined above consists of convex faces which have at most 8 edges on their boundary.*

Proof. Clearly, due to the packing and covering properties, the Voronoi cell of $l \in L$ properly contains a ball B^- of radius $k/2$ centered at l, and is contained in a ball B^+ of radius k centered at l. Hence the boundary of l's Voronoi cell must be contained in the annulus $B^+ \setminus B^-$. Consider two lines s and s' that are tangent to the Voronoi cell at points p and p', and intersect at a point from which the Voronoi cell is seen at an angle of 45°. Let t and t' be the tangents to B^- with the same slopes as s and s', and as close as possible to p and p', respectively. Clearly, p and l are on opposite sides of t (and likewise p' and t'). But any two points of t and t' inside $B^+ \setminus B^-$ have distance at least $k/4$, thus the Voronoi cell boundary between p and p' has to have length at least $\Delta \ge k/4$. Since due to a simple packing argument the number of Voronoi neighbors of l is bounded by 24, one of the Voronoi edges along the Voronoi cell boundary from p to p' has to be at least $\Delta/24 \ge k/96 \ge 3 + \epsilon$ long, for sufficiently small ϵ. The *turn angle* between two adjacent edges that bound some face of $\mathcal{CD}$ is equal to the angle between the supporting lines of their dual Voronoi edges. The Lemma follows from the fact that the angle between the supporting line of a long Voronoi edge and the supporting line of the next long Voronoi edge can be at most 45° by the above analysis, and the total turn angle for all pairs of consecutive edges that bound a single face of $\mathcal{CD}$ is exactly 360°. ∎

The above Lemma gives an intuition why we can hope to extract sufficiently many dual edges in the CDM. Essentially, if k is sufficiently large, there are many Voronoi edges which allow a "safe corridor" for a witnessing path between two landmarks l_1 and l_2. This is, of course, an overly pessimistic analysis, and in practice, k can be set much smaller (less than 10), still our simulation results show that many dual edges can be witnessed.

The following Observation suggests a heuristic for bootstrapping the embedding process:

OBSERVATION 2.1. *Let $l_1, l_2, l_3 \in L$ be three points which are all pairwise adjacent in the conservative Delaunay graph with $|l_i l_j| \le k\sqrt{3}$ for $i \ne j$. Then l_1, l_2, l_3 form a triangular face in the plane decomposition induced by the conservative Delaunay graph.*

That is, after exhibiting the CDM, we will be looking for three landmarks that are interconnected by short paths. They will give us the first face of our embedding of the CDM.

2.3 Embedding the Combinatorial Delaunay Map The embedding phase of the CDM is relatively simple. We first identify a small triangle in the CDM as suggested in Observation 2.1 and fix the position of the respective landmarks on the vertices of an equilateral triangle. We call these three landmarks *temporary perimeter nodes*. We then follow the exposition in [15] and determine the coordinates of the remaining "interior nodes" by an iterative procedure where each non-perimeter node v periodically updates its coordinates (x_v, y_v) as follows:

$$(2.1) \qquad \begin{bmatrix} x_v \\ y_v \end{bmatrix} = \frac{\sum_{w \text{ adjacent to } v} \begin{bmatrix} x_w \\ y_w \end{bmatrix}}{|\{w : w \text{ adjacent to } v\}|}$$

The fixed point of this process can also determined via solving a system of equations. However, the above iterative process allows for a distributed implementation. When the system has reached a stable state we examine the resulting embedding of the graph, and in particular, look for the *largest face cycle* in this embedding (in terms of the number of bounding edges), which typically corresponds to the face cycle of the outer face. We fix the nodes of this largest face cycle – which we now call *final perimeter nodes* – on the boundary of a square or unit circle and repeat the above procedure until all nodes have stabilized their position. These positions then define our final geometric embedding of the CDM.

Why do we believe that this procedure yields good results? On one hand, there is a theoretical guarantee; for 3-connected planar graphs, the combinatorial embedding is *unique* up to deciding which face to use as the outer face, that is the largest face determined after the first iterative procedure is really the outer face (if the latter has the longest face cycle of all faces in the embedding which is almost always the case in practice). The result of the second iterative procedure then computes the desired embedding with the correct face cycle being the outer boundary. Of course, in practice, the encountered CDMs are rarely 3-connected. So what we do in practice is the following: starting with a triangular face as perimeter we compute an embedding as above. In case of not 3-connected CDMs, the resulting embedding might exhibit many degeneracies (faces of area 0), but no crossings. We look for a face cycle of a non-degenerate face in the embedding, which is longer than the longest one found so far. Then we fix this cycle on the perimeter and repeat until the longest face cycle in the embedding is the outer cycle. Our simulations show that this simple heuristic solution delivers always very good results after few iterations, resembling faithfully the original network topology.

A few informal remarks about the distributed implementation. Even though the basic rubber-banding technique is decentralized, detecting a stable state and determining the longest face cycle are global operations. Even at the exact fixed point of (2.1), if faces of nearly zero area are present, global reasoning seems necessary to compute consistent face cycles. We leave these questions to future research.

Heuristically, one can make use of the Section 2.2, i.e. the fact that the "interior" faces (those that do not correspond to the holes or the surrounding area not covered by the deployment) are small. The boundaries of such faces can therefore be traced out by propagating "short-lived" messages ("time to live" parameter set to $20k$, say) using the "right hand rule" in the current embedding and detecting their return to the origin. These probes are rather quick and can be performed periodically. If a node is able to discover most of its adjacent faces in this manner, it can take this as a sign of near-optimal convergence and proceed to discover the remaining adjacent faces (some of which are potentially long) similarly, only using the "longer-range" messages; upon the first successful such long-range probe an announcement is broadcast to the network.

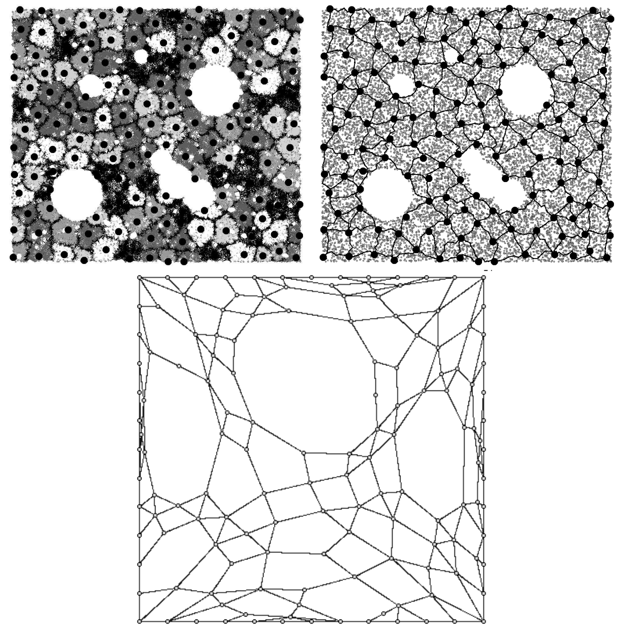

Figure 3: The three main steps of the construction of virtual coordinates: first the graph Voronoi diagram is computed; then paths are exhibited that witness adjacencies between tiles/landmarks maintaining planarity. Finally, after deriving part of the combinatorial embedding of the the resulting dual graph, we embed it using a simple "rubber-banding" scheme.

2.4 Summary In this section we have exhibited the main contribution of our paper. While computing a geometric realization of a complete unit-disk communication graph seems a very challenging problem in particular in a distributed scenario ([1, 9, 15]), taking a macroscopic view and concentrating on a subgraph of the combinatorial Delaunay graph, it turns out that a relatively simple, distributed algorithm can be used to extract a provably planar subgraph which we call *combinatorial Delaunay map*. Using a heuristic algorithm we can also compute a planar geometric embedding of the latter. Apart from nicely resembling the original network topology, this geometric planar embedding of the CDM allows the application of networking protocols that rely on the availability of geographic coordinates and a planar subgraph respecting these coordinates. See Figure 3 for an illustration of the main steps of our approach.

3 Application: Geographic Routing with Guaranteed Delivery in Location-unaware Sensor Networks

In *geographic routing protocols* the location of a node becomes its name or address, each node only needs to know the locations of its neighbors and the final destination in order to decide how to forward a packet. To guarantee delivery of messages within the network if the greedy forwarding phase gets stuck in a local minimum, geographic routing protocols like GPSR [8] not only rely on location information in the nodes but also on the identification of a planar subgraph to recover from such local minima. The appeal of these geographic forwarding methods is that they tend to produce routing paths close to best possible and the scalability even for very large networks. Virtual coordinates have been proposed to allow location-unaware networks the benefits of geographic routing; known constructions of such coordinates are rather demanding, and the identification of a suitable planar subgraph non-trivial, though.

One possible application of the embedding of the CDM that we have constructed is a *macroscopic geographic routing* protocol where geographic routing (like GPSR) is performed to determine the sequence of *tiles* to be traversed. As we have explicitly exhibited an embedding of a planar subgraph of the connectivity graph between the tiles, we can recover in case of local minima. Between the tiles, messages can be forwarded using local gradient fields, where each node in a tile knows the distances to all landmarks of adjacent tiles. We expect this method to perform very reliably and in particular efficiently on sparse network deployments due to the macroscopic view on the geometry. We are currently investigating the performance of this scheme in terms of load-balancing and quality of the produced routing paths.

4 Experimental Results

We first consider a large-scale deployment of reasonably densely connected nodes. Figure 4 shows that irregular (non-rectangular) outer boundary does not present a problem for our boundary detection method. It might only introduce additional distortion in the final embedding.

Figure 6 shows three stages of the process of finding the (approximate) outer face cycle. In this case it took only three invocations of the rubber-banding subroutine to arrive at the final solution. The network size is about 30000 and the average degree is 14. In all the examples that we tried the heuristic was successful in recovering a cycle which is very close to the true outer cycle in only a few (up to 5) iterations.

Our next experiment shows that having denser landmarks is not necessarily better. If the landmarks are too dense (Figure 7 top-right), very few CDM edges will be witnessed, since there will be a lot of "interference" between Voronoi cells. As a consequence, the CDM will be almost a forest (i.e. very weakly connected or even disconnected) and thus provide almost no useful information for the rubber-band embedding phase. The result will be a nearly trivial embedding, with a lot of overlapping landmarks. On the other hand, if landmarks are too sparse, the boundaries of smaller holes do not appear as long cycles in the embedding (Figure 7 bottom), so we lose sensitivity to small-scale features.

Finally, we test our algorithm on a very sparse deployment (Figure 5 top-left). One can see that the final embedding (Figure 5 bottom) is still a reasonably faithful representation in terms of topology. Note that since the faces produced by rubber-band embedding are always convex, our method tends to "convexify" the holes, as can be seen in the this example.

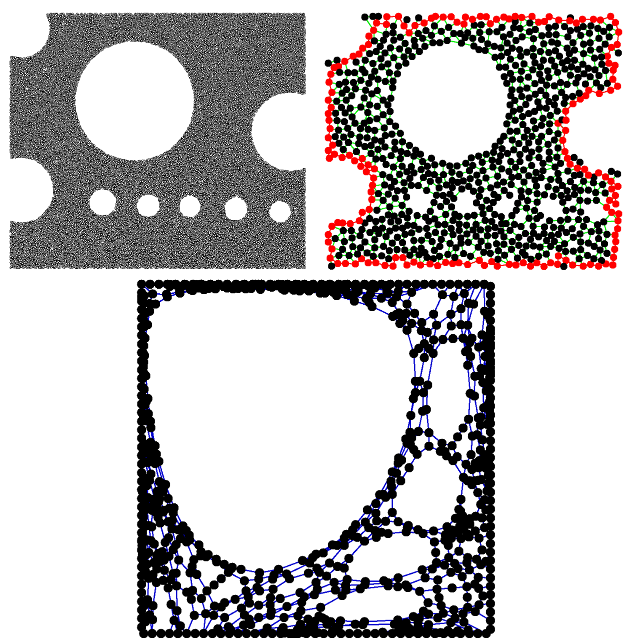

Figure 4: Performance on a large-scale deployment of 100000 nodes with average degree 12. Top-left: Node distribution (notice the irregular shape of the outer boundary). Top-right: The outer boundary cycle is correctly detected using $k = 3$. Bottom: The holes appear as large cycles in the final embedded CDM, with their relative sizes well preserved.

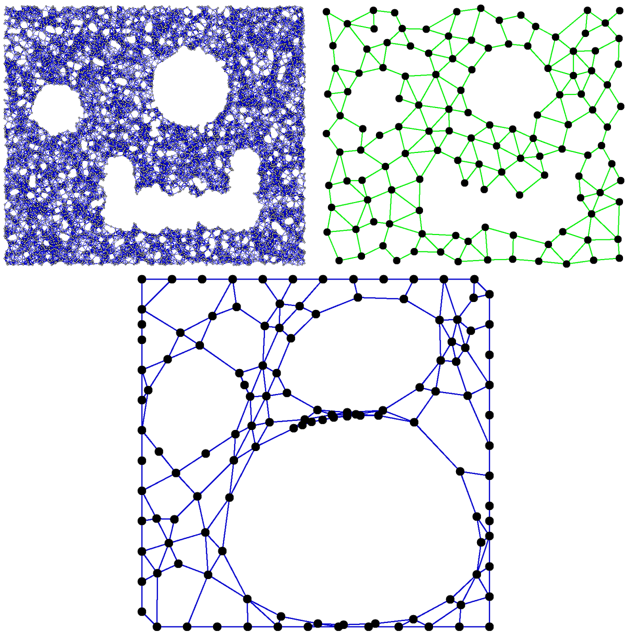

Figure 5: Performance on a sparse network. Top-left: A sparse communication graph, 9000 nodes with average degree 7; one can notice numerous small holes as a consequence of weak connectivity. Top-right: The CDM constructed with parameter $k = 4$. Bottom: The resulting embedded CDM.

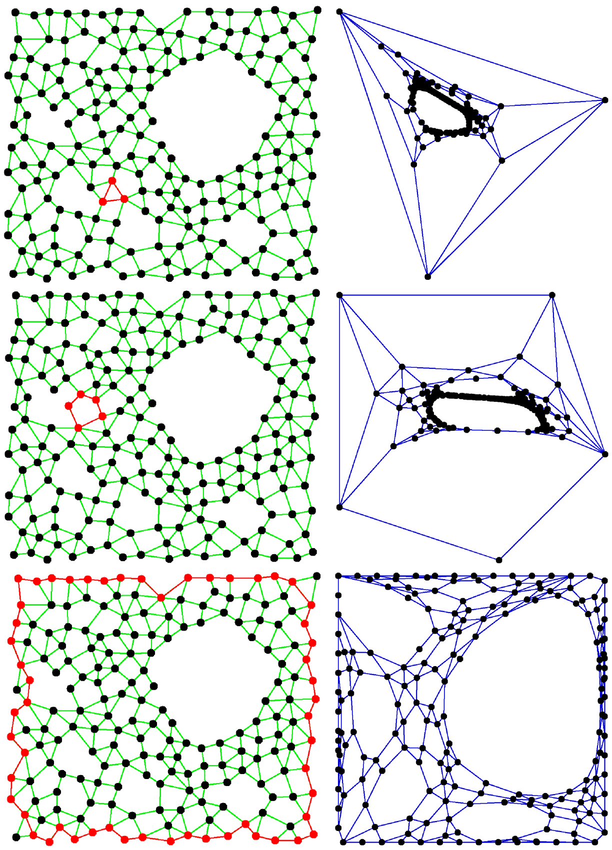

Figure 6: The three stages of the "boundary evolution" in our heuristic for finding the approximate outer face cycle. The right column shows a sequence of rubber-band embeddings, each of which is constructed using the cycle (shown in red, in the same row), and used to construct a new cycle (shown in the next row). The new cycle is always longer than the previous one.

5 Discussion

The tight connection between geographic location and connectivity of wireless networks gives rise to many interesting problems. While the connectivity graph of a wireless network does not explicitly hold any geometric information, the fact that its connectivity is determined by geographic proximity relations allows for techniques to extract (part) of the underlying geometry. This paper and few papers before have explored that direction but we believe that there are still many challenging problems to be solved. Ultimately, of course, one would like to recover the exact relative positions of the network nodes; unfortunately this problem is NP-hard for general unit-disk graphs as was shown by Kuhn et al. in [9], but a constant approximation has not been ruled out yet.

References

[1] Jehoshua Bruck, Jie Gao, and Anxiao (Andrew) Jiang. Localization and routing in sensor networks by local angle information. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 181–192, New York, NY, USA, 2005. ACM Press.

[2] Vin de Silva and Gunnar Carlsson. Topological estimation using witness complexes. In *Symposium on Point-Based Graphics*, June 2004.

[3] Vin de Silva and Robert Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *International Journal of Robotics Research*, December 2006. to appear.

[4] Martin Erwig. The graph voronoi diagram with applications. *Networks*, 36(3):156–163, 2000.

[5] S. P. Fekete, A. Kröller, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *Proc. of SODA*, 2006.

[6] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology

recognition in sensor networks. In *ALGOSENSORS*, 2004.

[7] Stefan Funke and Christian Klein. Hole detection or: "how much geometry hides in connectivity?". In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 377–385, New York, NY, USA, 2006. ACM Press.

[8] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243–254, 2000.

[9] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Unit disk graph approximation. In *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 2004.

[10] Fabian Kuhn and Aaron Zollinger. Ad-hoc networks beyond unit disk graphs. In *Proc. DIALM-POMC*, pages 69–78, 2003.

[11] M.F.Cox and M.A.A.Cox. *Multidimensional Scaling*. Chapman and Hall, 2001.

[12] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer. Virtual coordinates for ad hoc and sensor networks, 2004.

[13] Ilan Newman and Yuri Rabinovich. A lower bound on the distortion of embedding planar metrics into euclidean space. In *SCG '02: Proceedings of the eighteenth annual symposium on Computational geometry*, pages 94–96, New York, NY, USA, 2002. ACM Press.

[14] L. Guibas Q. Fang, J. Gao, V. de Silva, and L. Zhang. GLIDER: Gradient landmark-based distributed routing for sensor networks. In *24rd Conf. of the IEEE Communications Society (INFOCOM)*, 2005.

[15] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proc. MobiCom*, 2003.

[16] Satish Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 300–306, New York, NY, USA, 1999. ACM Press.

[17] Josh Tenenbaum, Vin de Silva, and John Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, (290):2319–2323, 200.

[18] Yue Wang, Jie Gao, and Joseph S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, September 2006.

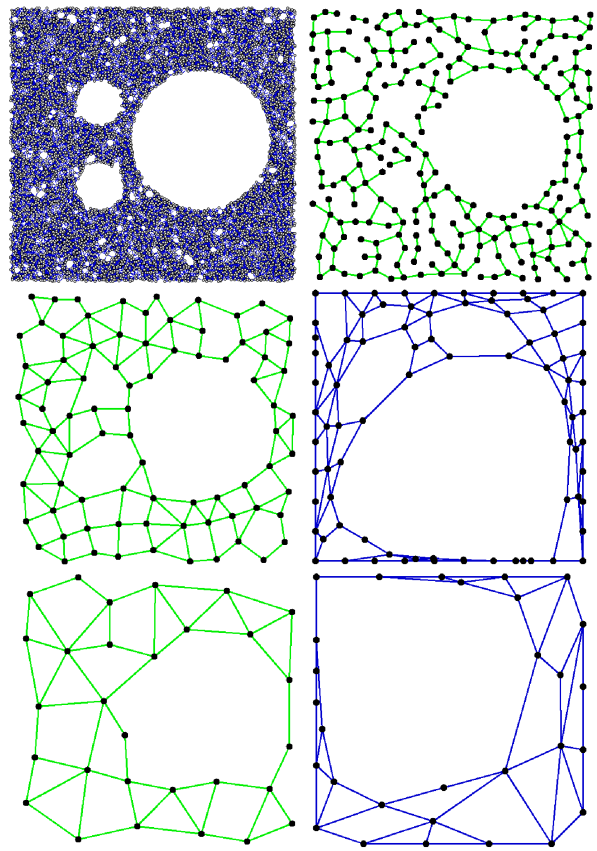

Figure 7: Effect of varying landmark density. Top-left: Communication graph with about 8400 nodes of average degree 10. Top-right: Too dense a set of landmarks ($k = 2$) may lead to poor results: the CDM has very few cycles that can be used for the embedding. Middle: An appropriate choice of landmark density ($k = 4$), both the larger and the two smaller holes show up in the embedding. Bottom: If the landmarks are too sparse ($k = 7$), the smaller holes become indistinguishable.