A Distributed Delaunay Triangulation Algorithm Based on Centroidal Voronoi Tessellation for Wireless Sensor Networks

Hongyu Zhou
zhou.hongyu@me.com

Miao Jin
mjin@cacs.louisiana.edu

Hongyi Wu
wu@cacs.louisiana.edu

The Center for Advanced Computer Studies (CACS)
University of Louisiana at Lafayette
Lafayette, LA 70503

ABSTRACT

A wireless sensor network can be represented by a graph. While the network graph is extremely useful, it often exhibits undesired irregularity. Therefore, special treatment of the graph is required by a variety of network algorithms and protocols. In particular, many geometry-oriented algorithms depend on a type of subgraph called *Delaunay triangulation*. However, when location information is unavailable, it is nontrivial to achieve Delaunay triangulation by using connectivity information only. The only connectivity-based algorithm available for Delaunay triangulation is built upon the property that the dual graph for a Voronoi diagram is a Delaunay triangulation. This approach, however, often fails in practical wireless sensor networks because the boundaries of Voronoi cells can be arbitrarily short in discrete sensor network settings. In a sensor network with connectivity information only, it is fundamentally unattainable to correctly judge neighboring cells when a Voronoi cell boundary is less than one hop. Consequently, the Voronoi diagram-based Delaunay triangulation fails. The proposed algorithm employs a distributed approach to perform centroidal Voronoi tessellation, and constructs its dual graph to yield Delaunay triangulation. It exhibits several distinctive properties. First, it eliminates the problem due to short cell boundaries and thus effectively avoids crossing edges. Second, the proposed algorithm is proven to converge and succeed in constructing a Delaunay triangulation, if the CVT cell size is greater than a constant threshold. Third, the established Delaunay triangulation consists of close-to-equilateral triangles, benefiting a range of applications such as geometric routing, localization, coverage, segmentation, and data storage and processing. Extensive simulations are carried out under various 2D network models to evaluate the effectiveness and efficiency of the proposed CVT-based triangulation algorithm.

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer-Communication Networks—*Network Architecture and Design*

MobiHoc'13, July 29–August 1, 2013, Bangalore, India.

Keywords

Delaunay, Triangulation, Centroidal Voronoi Tessellation, Wireless Sensor Networks

1. INTRODUCTION

This work aims to develop a distributed and efficient algorithm to construct Delaunay triangulation for wireless sensor networks. This section first introduces the motivations, then discusses the state-of-the-art Delaunay triangulation algorithms, followed by a summary of the main contributions of this paper.

1.1 Motivation

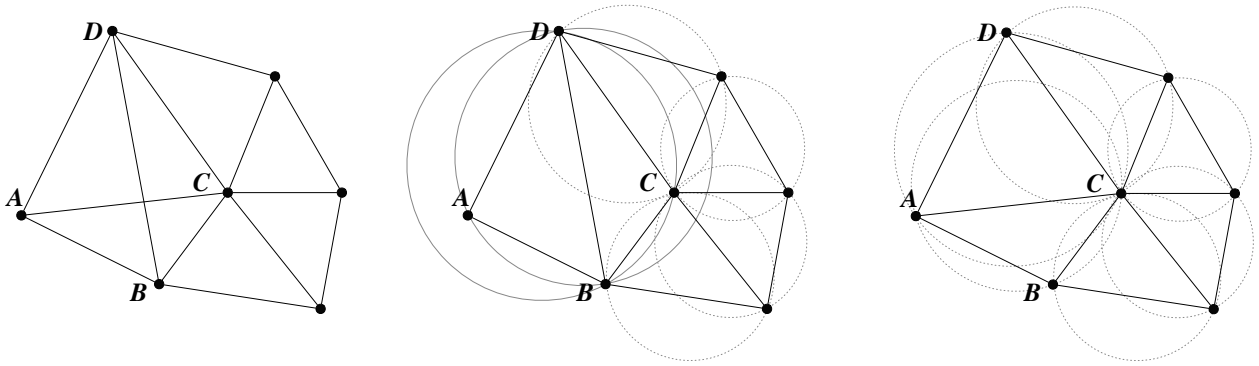
A wireless sensor network can be represented by a graph, where a node corresponds to a sensor and an edge indicates the communication link between two adjacent sensors with radio transmission range. While the network graph itself is extremely useful, it often exhibits undesired randomness and irregularity. Therefore, special treatment of the graph is required by a variety of network algorithms and protocols. In particular, many geometry-oriented algorithms, such as geometric routing [1–4], autonomous localization [5, 6], sensor coverage [7], network segmentation [8], and distributed data storage and processing [9], all depend on a type of subgraph called *triangulation*.

In general, triangulation is a subdivision of a geometric object into simplices. The triangulation of a discrete set of points is a subdivision of the convex hull of the points into simplices such that any two simplices intersect in no more than one common face and the vertices of the subdividing simplices coincide with the points [10]. Several algorithms have been proposed for triangulation in wireless sensor networks [1, 8, 11]. Obviously, while a network graph is usually nonplanar (see the crossing edges in Fig. 1(a)), a triangulation is a planar graph, where no edges cross each other (as illustrated in Fig. 1(b)).

Delaunay triangulation is a special triangulation defined as follows.

Definition 1. *A Delaunay triangulation for a set of points on a plane is a triangulation such that no point is inside the circumcircle of any triangle [12].*

An example of Delaunay triangulation is shown in Fig. 1(c), where none of the circumcircles of the triangles contain a node. On the other hand, the triangulation in Fig. 1(b) does not meet the above definition. For instance, the circumcircle of $\triangle ABD$ contains Node C. Thus it is not a Delaunay triangulation. The Delaunay triangulation is preferred in many network algorithms, because it



(a) A nonplanar network graph. (b) A triangulation of the graph. (c) A Delaunay triangulation of the graph.

Figure 1: Triangulation and Delaunay triangulation of a nonplanar network graph.

tends to avoid skinny triangles by maximizing the minimum angle in the triangulation. For instance, greedy forwarding is guaranteed to succeed in a Delaunay triangulation except at network boundaries [1–4]. And better localization can be achieved based on a Delaunay triangulation than that based on non-Delaunay triangulation [5, 6]. The Delaunay triangulation with all edges equal is call an equilateral Delaunay triangulation.

1.2 State-of-the-Art for Connectivity-Based Delaunay Triangulation

If the location information or distance measurement is available, a Delaunay triangulation can be straightforwardly constructed [4]. However, it is nontrivial to achieve Delaunay triangulation by using connectivity information only. Note that the triangulation method proposed in [1, 8] is not Delaunay-guaranteed.

The only connectivity-based algorithm for Delaunay triangulation in practical wireless sensor networks is built upon the property that the dual graph for a Voronoi diagram[1] is a Delaunay triangulation. To this end the planarization algorithm proposed in [13] is employed to establish an approximate Voronoi diagram. More specifically, a node is randomly chosen as a generating point, which claims its K-hop neighbors to form a cell. Then a node is randomly chosen among the rest nodes as the next generating point. The process repeats until every node in the network is either selected as a generating point or associated with a generating point. If a node is claimed by multiple generating points, it chooses the closest one (in term of hop count) and joins its cell. The distance between any two adjacent generating points is greater than K hops but no more than $2K+1$ hops.

Then, the dual graph of the approximate Voronoi diagram, called combinatorial Delaunay graph (CDG), is constructed as follows. If two cells are adjacent to each other, i.e., have at least one pair of neighboring nodes (one in each cell), a virtual edge is established to connect the corresponding generating points. Note that, while the precisely computed dual graph of a Voronoi diagram is a Delaunay triangulation as discussed earlier (e.g., as shown in Fig. 2(a)), the same result no longer holds for CDG. As a matter of fact, CDG is not even necessarily planar, as illustrated in Fig. 2(b) that shows crossing edges. This is because the boundary of two adjacent approximate Voronoi cells can be shorter than one hop, and thus two non-neighboring cells may be mistakenly considered as neighbors. For example, as illustrated in Fig. 2(b), the shared boundary of Cells A and B is virtually zero (with only one pair of nodes connected). At the same time, a node in Cell C is directly connected to a node in Cell D. Hence Cells C and D are deemed adjacent, resulting in crossing edges. Practically, there are less crossing edges under a larger K. However no matter how large the K (i.e., the cell) is, the approximate Voronoi diagram cannot guarantee every cell boundary to be greater than one hop. Therefore this scheme does not ensure the success of Delaunay triangulation. Moreover, even a Delaunay triangulation is successfully constructed, the formed triangles are often nonuniform in terms of edge length (see Fig. 2(a)).

The crossing edges in CDG can be removed, yielding a Combinatorial Delaunay Map (CDM), which is planar but has polygon holes [13]. Therefore it is not a triangulation. The algorithms proposed in [1, 11] try to fill the holes to form triangles. However, the resulting triangulation is not guaranteed to hold the Delaunay property.

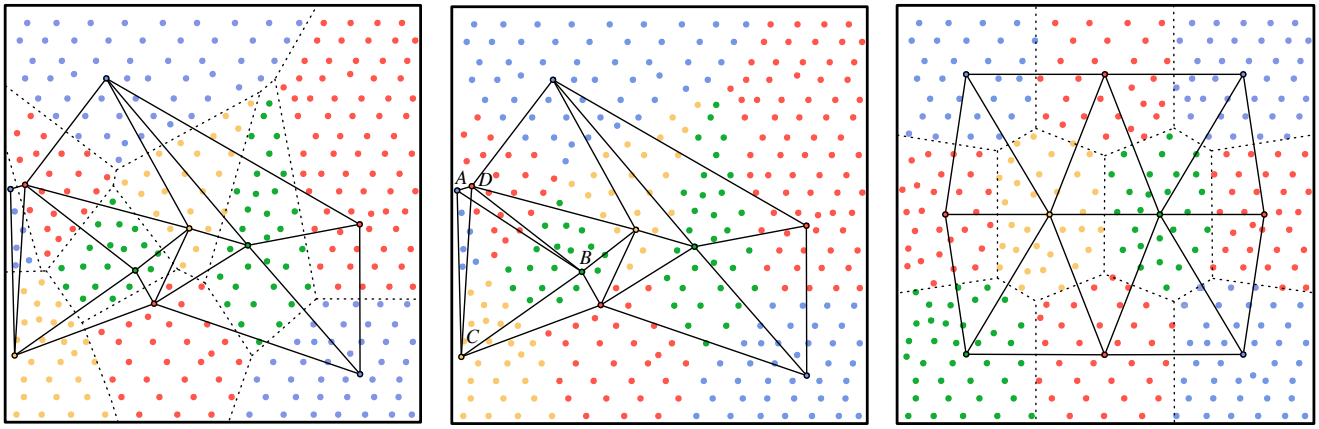
[1] Given a finite set of generating points on a plane, the Voronoi diagram is a partitioning of the plane with points into convex polygons (or cells) such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other [12].

1.3 Contribution of This Work

The problem of the Voronoi diagram-based (or VD-based) algorithm stems from the fact that the boundaries of Voronoi cells are nonuniform. Particularly, some cell boundaries can be arbitrarily short. In a sensor network with connectivity information only, it is fundamentally unattainable to correctly judge neighboring cells when a Voronoi cell boundary is less than one hop. Consequently, some generating points are mistakenly connected, forming crossing edges. As a result, the VD-based Delaunay triangulation fails. It is worth clarifying that Delaunay triangulation can aways be achieved if accurate location information is available. With the location information, a Voronoi diagram can be precisely computed, and accordingly its dual graph can be readily obtained by connecting the generating points whose cells share a boundary (no matter how short the boundary is). Such a dual graph is proven to be a Delaunay triangulation.

It is obviously desired to build a Voronoi diagram with uniform cell boundaries, so as to maximize the minimum boundary length. This observation motivates a new triangulation algorithm based on centroidal Voronoi tessellation (CVT).

Definition 2. *A centroidal Voronoi tessellation (CVT) is a special Voronoi diagram, where the generating point of each Voronoi cell is also its mean (i.e., center of mass) [14].*



(a) (b) (c)

Figure 2: Comparison between the dual graphs of Voronoi diagram and CVT. (a) The dual graph of a precisely computed Voronoi diagram. (b) The dual graph with crossing edges of an approximate Voronoi diagram under discrete sensor network settings. (c) CVT has uniform cell boundaries and thus its dual graph forms a Delaunay triangulation with close-to-equilateral triangles.

The proposed algorithm employs a distributed approach to perform centroidal Voronoi tessellation, and constructs its dual graph to yield Delaunay triangulation. It exhibits several distinctive properties. First, it eliminates the problem due to short cell boundaries and thus effectively avoids crossing edges, significantly improving the probability to successfully establish a Delaunay triangulation in comparison with its VD-based counterpart. Second, as shown by Theorems 1 and 2, the proposed algorithm can always converge and succeed in constructing a Delaunay triangulation, if the CVT cell size is greater than a constant threshold. Third, the established Delaunay triangulation consists of close-to-equilateral triangles, benefiting a range of applications such as geometric routing [1–4], autonomous localization [5,6], sensor coverage [7], network segmentation [8], and distributed data storage and processing [9].

The rest of this paper is organized as follows: Sec. 2 introduces the proposed CVT-based Delaunay triangulation algorithm and proves its convergence and correctness when the cell size is greater than a constant threshold. Sec. 3 presents simulation results. Finally, Sec. 4 concludes the paper.

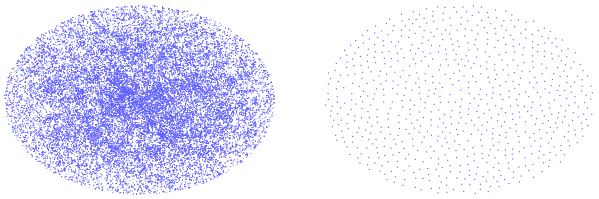
2. PROPOSED CVT-BASED TRIANGULATION ALGORITHM

This section presents the proposed CVT-based triangulation algorithm and proves its convergence and correctness when the cell size is greater than a constant threshold.

The proposed algorithm consists of three phases. First, it samples the generally distributed network to yield a uniform nodal density. Second, it iteratively builds an approximate CVT. Finally it obtains the dual graph of the CVT to construct the Delaunay triangulation.

2.1 Single-Hop Voronoi Diagram Sampling

As shown in Definition 2, the generating point of a CVT cell is the centroid of the cell. In order to build uniform cells, which induce a Delaunay triangulation with equilateral triangles, the network must have a uniform density function [14]. However, a general sensor network can be non-uniformly distributed, due to the lack of precise nodal deployment and the nondeterministic sensor failures and link dynamics. To this end, a sampling process is employed. It essentially builds a Voronoi diagram with small, constant cell size (e.g., one hop). The generating points of the Voronoi diagram have a uniform density, and thus serving as the ideal input for CVT construction.



(a) Non-uniformed topology. (b) Sampled subgraph.

Figure 3: Single-hop Voronoi diagram sampling that yields uniform nodal density.

More specifically, let an undirected graph $G = \{V, E\}$ represent a wireless sensor network, where V is the set of sensor nodes and E is the set of communication links. A node can be in one of the three possible states, i.e., a cell generating point, a cell member, or undetermined. Each node is initialized as the undetermined state. It starts a random back-off timer. If it is in the undetermined state when the timer expires, it changes its state to a generating point, and informs its undetermined one-hop neighbors to change their states to cell member. The process terminates when every node in the network becomes either a generating point or a cell member. Two generating points are adjacent if a pair of their cell members (one in each cell) are one-hop neighbors. The generating points are largely uniformly distributed, where any two adjacent generating points are at least one hop and at most three hops away from each other. Let $G' = \{V', E'\}$ be the sampled subgraph, where V' consists of the generating points and E' are the virtual links that connect adjacent generating points.

An example of sampling is shown in Fig. 3(b), where the nodes are uniform, in comparison with the nonuniform original graph Fig. 3(a).

2.2 CVT Construction

Based on the sampled subgraph G', the CVT is constructed via a distributed, iterative process. It starts with an arbitrary Voronoi di-

agram with a cell size of K-hops, which can be established according to the method introduced in Sec. 1.2. Let $T = \{T_1, T_2, ..., T_m\}$ denote the set of tessellations (or cells) and $L = \{L_1, L_2, ..., L_m\}$ the corresponding generating points of the initial Voronoi diagram. They will be updated by repeating the following two steps.

(1) Cells construction: The nodes in G' are associated with their closest generating points to form tessellations. Given Node n_i, its closest generating point is

$$L(n_i) = \arg\min_{L_j \in L} D(n_i, L_j), \tag{1}$$

where $D(n_i, L_j)$ denotes the hop distance between n_i to L_j. The nodes associated with the same generating point form a tessellation $T_j = \{n_i \in G' | L(n_i) = L_j\}$. Let $|T_j|$ denote the number of nodes in T_j. The colored cells in Fig. 4(a) show the initial tessellation.

(2) Centroid Calculation: The generating point of each cell is updated to the current centroid of the cell. More specifically, every node learns its hop distance to every other node in the same cell via localized flooding (within its cell). Then each node calculates the standard deviation of such distances. For example, given Node n_i in Cell T_j, its standard deviation σ_i is determined by the following equations:

$$\sigma_i = \sqrt{\frac{1}{|T_j| - 1} \sum_{n_k \in T_j} (D(n_i, n_k) - \overline{D_j})^2}, \tag{2}$$

where $\overline{D_j}$ is the mean of distances between any two nodes in Cell T_j, i.e.,

$$\overline{D_j} = \frac{2}{|T_j|(|T_j| - 1)} \sum_{n_p, n_q \in T_j} D(n_p, n_q). \tag{3}$$

The node with the minimal standard deviation is the approximate centroid of the tessellation and selected as the new generating point, i.e,

$$L_j = \arg\min_{n_k \in T_j} \sigma_k. \tag{4}$$

The above calculation involves all nodes in the cell (i.e., T_j). In fact, given the uniformly distributed nodes (after sampling), the centroid can be determined according to the cell boundary that depicts the shape of the cell. The boundary nodes of the cell can be easily identified. If a node is on the network boundary or has a neighbor node that belongs to a different cell, it is a cell boundary node. The centroid calculation can be carried out with reduced computing time, communication overhead and energy consumption by simply replacing T_j in Eqs. (2)-(4) with the set of cell boundary nodes. The boundary-based approximation yields similar CVT construction as to be demonstrated in Sec. 3.

Once the new generating points are determined, the cells are updated accordingly as discussed in the previous step.

Fig. 4 illustrates the evolution of the cells while the above steps repeat. As can be seen, the cells become more uniform after more iterations. The process terminates when the generating points remain unchanged. Fig. 4(d) shows the final CVT.

The iterative process always converges as shown by the following theorem.

Theorem 1. *The proposed CVT construction algorithm converges.*

PROOF. For a given cell j, both $|T_j|$ and $\overline{D_j}$ are constant. Thus, CVT construction intrinsically aims to minimize the following optimal function $J(T, L)$:

$$J(T, L) = \sum_{i \in G'} (D(n_i, L(n_i)))^2. \tag{5}$$

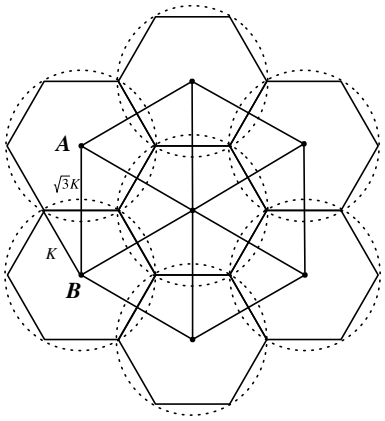


Figure 5: An ideal CVT consists of regular hexagons.

Assume J is not minimal, one can either fix generating points L to adjust the nodes' association to reduce $J(T, L)$ just like cells construction, or fix cell nodes C to select new generating points as the centroids. Both processes are monotonically decreasing $J(T, L)$. When $J(T, L)$ reaches the minimal value, the cells T and generating points L converge at the same time. As a matter of fact, the CVT construction algorithm is a coordinate descent algorithm with guaranteed convergence. □

While it is difficult to derive a theoretic bound for the number iterations in order to reach the convergence, the algorithm converges very fast in practice. For example, merely four to five iterations are needed in most cases according to the simulation results to be presented in Sec. 3.

2.3 Delaunay Triangulation

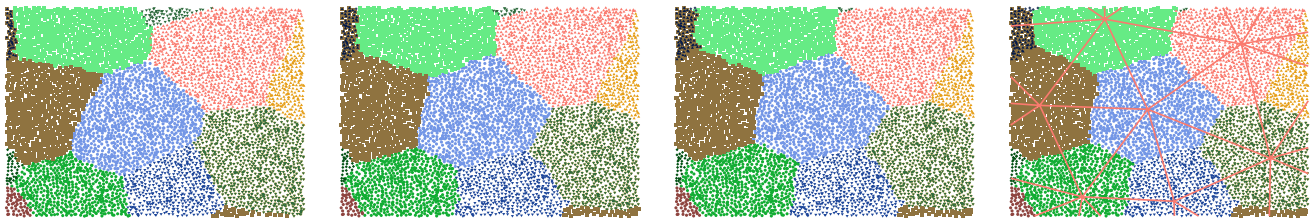
After CVT is constructed, its dual graph is established by connecting every two adjacent generating points with a virtual edge. Two generating points are adjacent if at least one pair of their cell members (one in each cell) are neighbors in G'. An example of the dual graph is illustrated Fig. 4(d).

The following theorem formally shows that the proposed algorithm can always construct a Delaunay triangle mesh, if the cell size is greater than a constant threshold. Without loss of generality, the maximum radio transmission range is normalized to one in the following discussions.

Theorem 2. *Given an asymptotically deployed wireless sensor network, the centroidal Voronoi tessellation (CVT) with a cell size greater than 15 always yields a Delaunay triangulation.*

PROOF. As introduced in Sec. 2.1, a sampling process (that results in a constant density function) is employed before the centroidal Voronoi tessellation. Accordingly, every cell of CVT is a regular hexagon (see Fig. 5) if the nodal density approaches infinity (i.e., the continuous case) and the network is asymptotically deployed with no boundaries [15]. The distance between any two neighboring centroids is $\sqrt{3}K$, where K is the hexagon edge length or the cell radius. The dual of the CVT is a Delaunay triangle mesh that consists of equilateral triangles [14]. For convenience, such a CVT is referred as the *ideal CVT* and the corresponding Delaunay triangle mesh is referred as the *ideal triangulation* in the following discussions.

Under a discrete sensor network setting, the proposed algorithm yields an approximate CVT, where the tessellation is performed according to hop counts instead of real distance. Thus, the actual cell



(a) Initial Voronoi diagram. (b) Result after two iterations. (c) Result after four iterations. (d) Final result.

Figure 4: CVT-Based Delaunay Triangulation Algorithm. It starts with an arbitrary Voronoi diagram (depicted in (a)), and refines the cells by a small number of iteration (see (b) and (c)) to yield the final CVT cells (shown in (d)). The dual graph of the CVT cells is a close-to-equilateral Delaunay triangulation.

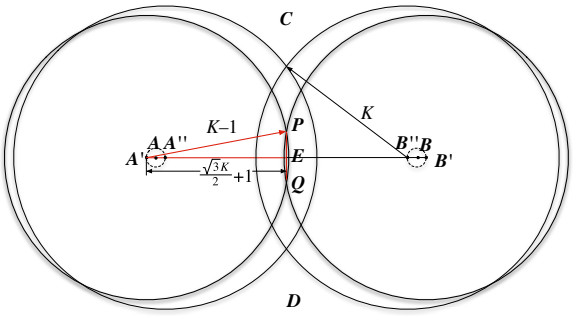


Figure 6: Illustration for the proof of Theorem 2. The length of PQ reaches minimum when the two generating points are farthest away from each other (i.e., at A' and B') and the cell size is the smallest (i.e., $K-1$).

size is in the range of $(K-1, K]$. Given such an approximate CVT cell, a node is not always available at its real centroid. For example, Fig. 6 illustrates two adjacent cells, with their real centroids at A and B, respectively. The actual generating point of a cell could be anywhere within 1-hop range of the real centroid, as delineated by the small dashed circle around A or B. Similar to the continuous setting, the proposed algorithm obtains the dual of the approximate CVT, aiming to produce a triangle mesh.

The rest of the proof shows that, when K is greater than a constant, the approximate CVT induces the same triangle mesh as the ideal triangulation, with neither extra edges nor missing edges.

First, if an extra edge is added into the ideal triangulation, it must result in a crossing edge [12]. To ensure free of crossing edges in the dual of the approximate CVT, the boundary between the two cells must be greater than the radio transmission range (i.e., 1). Otherwise, the cells on the two sides of AB (i.e., Cells C and D) might be mistakenly considered as neighboring cells since their nodes can be connected, leading to a crossing edge CD (see a similar example in Fig. 2(b)). To this end, the proof intends to examine the worst case scenario that results in the shortest boundary between Cells A and B. The worst case occurs when the two generating points are farthest away from each other (i.e., at A' and B' in Fig. 6) and the cell size reaches minimum (i.e., $K-1$). Let P and Q denote the intersections of the two cells under the worst case, and E denote the intersection of AB and PQ. Obviously, $\|B'E\| = \sqrt{3}K/2+1$ and $\|B'P\| = K-1$. Let the boundary greater than one, i.e.,

$$\|PQ\| = 2\sqrt{\|B'P\|^2 - \|B'E\|^2} > 1. \qquad (6)$$

The inequality holds when and only when K is greater than 15.06. Therefore, if $K > 15$, the dual graph of the approximate CVT includes no extra edges compared with the ideal triangulation.

Second, since PQ is greater than one, Cells C and D are disconnected, i.e., there does not exist a path between C and D that involves the nodes in the two cells only. If Cells A and B are not connected either, there must exist a void (i.e., a hole) in the middle of the four cells. The hole forms a network boundary [16], which contradicts to the assumption of asymptotic sensor deployment with no boundaries. Hence, Cells A and B must be neighbors, inducing Edge AB in the dual of the approximate CVT.

Therefore, if $K > 15$, the dual graph of the approximate CVT is the same as the ideal triangulation, i.e., a Delaunay triangle mesh. □

Note that Theorem 2 only shows a provable sufficient condition for Delaunay triangulation. It is not always necessary to have $K > 15$ to establish a Delaunay triangle mesh. When $K \leq 15$, although without a proof, it is intuitively obvious that the proposed algorithm yields better triangulation in comparison with its VD-based counterpart. More specifically, it results in close-to-equilateral triangles. The uniform edge length reduces the probability of crossing edges as discussed in Sec. 1. This observation is verified by simulation results to be presented in the next section.

2.4 Time Complexity and Communication Cost

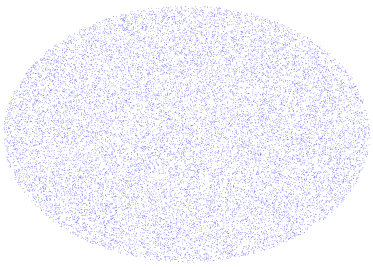
The proposed Delaunay triangulation algorithm has a linear time complexity and communication cost (measured by messages sent) with respect to the size of the network. A brief analysis is summarized below.

First, the single-hop Voronoi diagram sampling has a time complexity of $O(n)$, where n is the total number of nodes in the network. As each node communicates with its neighbors only, the communication cost is also $O(n)$.

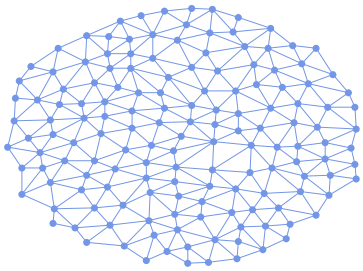
In every iteration of CVT construction, each cell has a time complexity and communication cost of $O(m^2)$, where m is the number nodes in a cell. Since all cells are processed simultaneously, the network-wide time complexity remains $O(m^2)$, but the overall communication cost becomes $O(cm^2)$, where c is the number of cells. Obviously, $O(cm) = O(n)$, hence $O(cm^2) = O(nm)$. For a given cell size (i.e., K) and nodal density, m is bounded by a constant. Therefore, $O(nm) = O(n)$. Moreover, the proposed algorithm adopts a constant number of iterations (e.g., four iterations that are enough to yield satisfied results).

Finally, the time complexity and communication cost for dual graph construction are both $O(k)$.

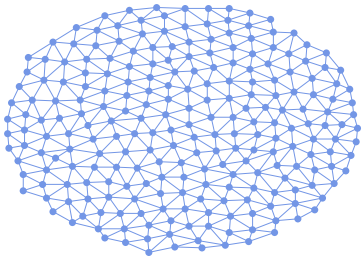
In summary, the overall time complexity and communication cost of the Delaunay triangulation algorithm are dominated by $O(n)$.



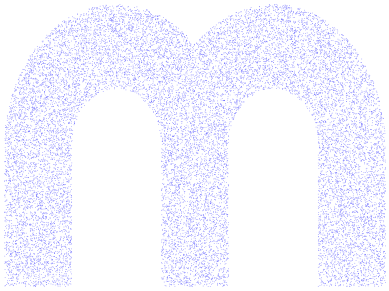
(a) An oval-shape network.



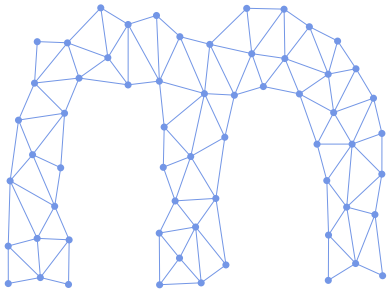
(b) Finest triangulation by VD ($k = 15$).



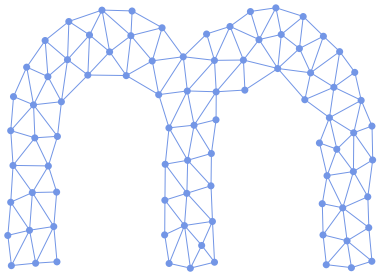
(c) Finest triangulation by CVT ($k = 12$).



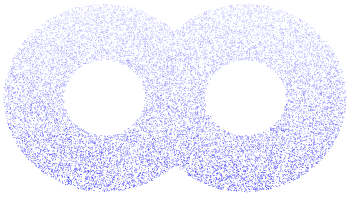
(d) An m-shape network.



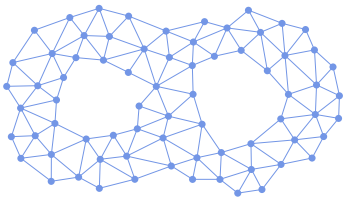
(e) Finest triangulation by VD ($k = 16$).



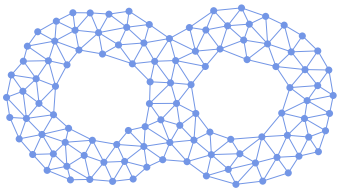
(f) Finest triangulation by CVT ($k = 13$).



(g) An 8-shape network (with 2 holes).



(h) Finest triangulation by VD ($k = 15$).



(i) Finest triangulation by CVT($k = 12$).

Figure 7: Example of 2D networks that demonstrate the proposed CVT-based algorithm produces finer triangulations than the VD-based algorithm does.

3. SIMULATIONS RESULTS

In order to evaluate the effectiveness and efficiency of the proposed CVT-based triangulation algorithm, extensive simulations have been carried out under various 2D network models. The sensor nodes are randomly deployed in each network. The proposed algorithm does not depend on any specific communication model. This simulation adopts a general model, with merely a constraint on the maximum radio transmission range, which is normalized to one. It is similar to the most general Quasi-UDG model. Quasi-UDG determines connectivity according to a parameter $\alpha < 1$. Two nodes are disconnected if they are separated by a distance greater than one, or connected if their distance is less than α, or connected with a probability if their distance is between α and one. In this simulation, α is set to 0. The performance of the proposed CVT-based algorithm is compared against its VD-based counterpart, in terms of crossing edge rate, the deviation of triangle edge length, and the success rate of Delaunay triangulation construction.

3.1 Triangulation Granularity

First, simulations are performed to compare the granularity of triangulations produced by the proposed CVT-based algorithm and its VD-based counterpart. Several sample networks are illustrated in Fig. 7, in oval, m-shape, and 8-shape, respectively. As discussed in Sec. 2, the cell size K is crucial for both algorithms, dictating the possibility of having crossing edges. K must be large enough to ensure the free of crossing edge, and thus a valid Delaunay triangulation.

The simulation intends to find the minimum K under both algorithms, which induce a successful triangulation. A small K is highly desired because it represents the original network with finer granularity. The simulation results show that the CVT-based algorithm always yields a finer triangulation (with smaller K) than the VD-based algorithm does (see the second and the third columns of Fig. 7).

As proven in Sec. 2, CVT with a cell size greater than $K = 15$ ensures a Delaunay triangulation. Note that, $K = 15$ is the theoretical bound. A small K is often sufficient in practice. However, the VD-based algorithm does not have such nice property. As a matter of fact, it is interesting to observe that even when the VD-based algorithm successfully produces a Delaunay triangulation under certain K, it may still fail when K further increases. This is evidenced in Fig. 8(a) that shows the crossing edge rate. The crossing edges are the edges that intersect, i.e., are non-planar. The crossing edge rate is defined to be the ratio between the number of crossing edges to

the total number of edges in the dual graph of VD or CVT. Obviously, the smaller the crossing edge rate, the better. A successful Delaunay triangulation is constructed if the crossing edge rate is zero. As can be seen, the crossing edge rate reaches and stays to be zero after $K = 12$ under the CVT-based algorithm. The VD-based algorithm, however, offers no guarantee for achieving free of crossing edges. The crossing edge rate exhibits significant fluctuations even when K is large.

It is also observed in simulation that the boundary will have some impact on the CVT construction. The shape of the boundary cells are more likely irregular (non-hexagon) compared with inner cells, because the shape of boundary cells are constrained by the network shape. Crossing edges occur more often around boundary cells.

3.2 Triangulation Regularity

In addition to granularity, it is obvious that the triangulation produced by the CVT-based algorithm is more uniform. This is because the CVT cells are generally more regular and uniform. Fig. 8(b) compares the deviation of cell size under the CVT and VD-based algorithms. As can be seen, the former achieves a consistently lower deviation in cell size than the latter. This is a natural result of CVT cell construction, which intends to improve the regularity of cells over the Voronoi diagram. Consequently, the edges of the dual graph under the former always have a smaller deviation than that of the latter.

The regularity of triangulation benefits a range of applications such as geometric routing, localization, coverage, segmentation, and data storage and processing. For example a connectivity-based localization method is introduced in [6]. It takes triangulations as input and applies a Ricci flow algorithm to computes the optimal flat metric of the triangulation in order to embed the network to plane (and accordingly determining the locations of sensors). It has been shown in [6] that the algorithm achieves the highest localization accuracy compared with other competing methods including multi-dimensional scaling (MDS) [17, 18] and neural network based methods [19, 20] under various representative network shapes.

This simulation intends to demonstrate the benefits of CVT-based triangulation in support of localization. To this end, the same cell size (i.e., K) is employed to build both VD and CVT cells to construct triangulations. Two network examples are depicted in Fig. 9, where nodes are randomly deployed with representative boundary shapes. The triangulations serve as inputs for the same localization algorithm. The localization result are shown in the second row of Fig. 9, where a blue line segment is drawn for each node, starting from its real coordinates marked with blue dot and ending at the computed coordinates. The gray triangulation is based on the computed coordinates. The longer the line segment, the lower accuracy of the localization results. It is obviously noticed that the CVT-based triangulation helps yield more accurate localization results. The quantitative average localization errors are summarized in Table 1. The localization error is computed as the ratio of the average node distance error (based on all edges in the network) and the averaged transmission range. As can be seen, a significant reduction of error (up to 50%) is achieved by using the CVT-based triangulation.

3.3 Comparison Cell Centroid Calculation

As discussed in Sec. 2, the calculation of cell centroid can be performed according to all nodes in the cell or only boundary nodes of the cell. Fig. 10 compares their performance. As can be seen, they achieve similar triangulation results. However, the latter involves only 24% of the nodes in comparison with the former, thus significantly reducing the communication overhead and energy consumption.

Table 1: Localization errors of [6] with CVT-based and VD-based triangulations as inputs.

	C-shape Model (see Fig. 9(a)-9(b))	Rectangle Model (see Fig. 9(c)-9(d))
VD	0.46	0.24
CVT	0.21	0.18

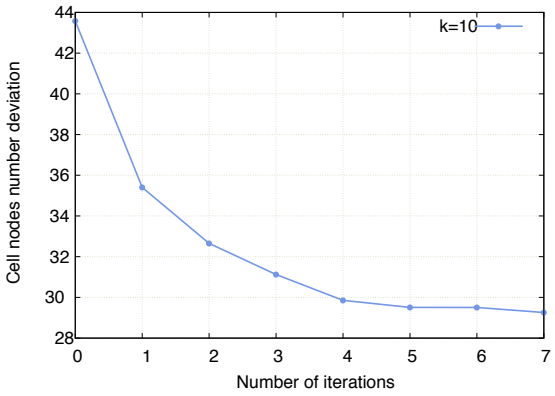


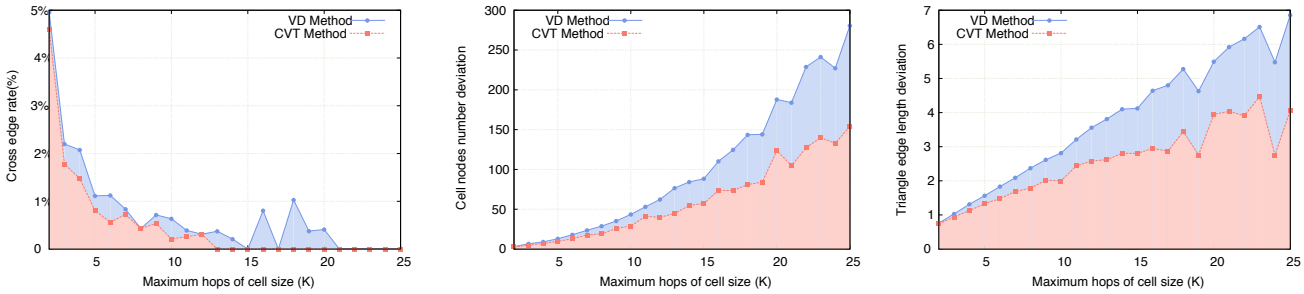
Figure 11: CVT triangulation convergence (K=10).

3.4 Convergence

The convergence of the proposed algorithm has been proven in Sec. 2. Fig. 11 shows the convergence speed under practical implementation. The standard deviation of cell size reaches minimum and keeps constant after about five iterations. The fast convergence is highly desired, because it means not only shorter algorithm running delay but also lower communication overhead. In each iteration, every nodes in the network must communicate with other nodes in the same cell to determine the new generating point and construct the CVT cell. As discussed in Sec. 2, the overall communication cost of one iteration is $O(nm)$, where m is the number nodes in a cell (which can be deem as a constant for a given K) and n is the number of nodes in the entire network. Apparently, a smaller number of iterations can significantly reduce the communication overhead and energy consumption in practical sensor network settings. The result also justifies the assumption of constant number of iterations used in complexity analysis in Sec. 2.

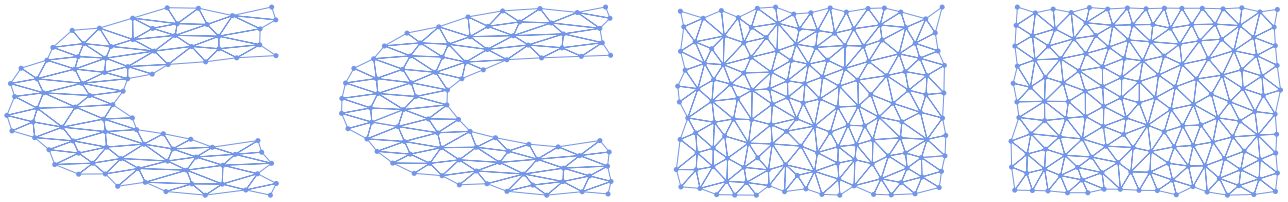
4. CONCLUSION

A wireless sensor network can be represented by a graph. While the network graph is extremely useful, it often exhibits undesired randomness and irregularity. Therefore, special treatment of the graph is required by a variety of network algorithms and protocols. In particular, many geometry-oriented algorithms depend on a type of subgraph called *Delaunay triangulation*. However, when location information is unavailable, it is nontrivial to achieve Delaunay triangulation by using connectivity information only. The only connectivity-based algorithm available for Delaunay triangulation is built upon the property that the dual graph for a Voronoi diagram is a Delaunay triangulation. This approach, however, often fails in practical wireless sensor networks because the boundaries of Voronoi cells can be arbitrarily short in discrete sensor network settings. In a sensor network with connectivity information only, it

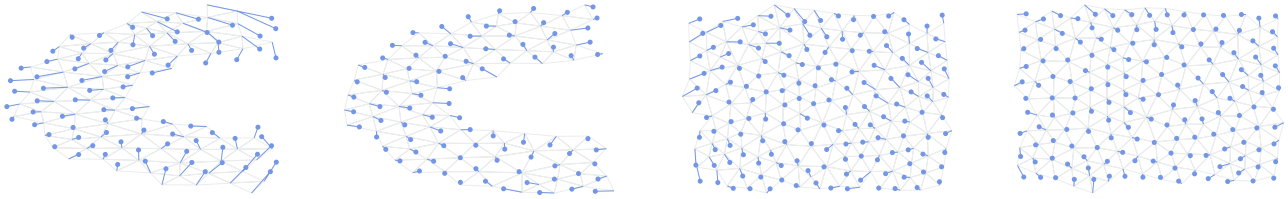


(a) Crossing edge rate. (b) Deviation of cell size. (c) Deviation of triangle edge length.

Figure 8: Quantitative comparison of CVT-based and VD-based triangulation algorithms. As depicted in (a), the crossing edge rate reaches and stays to be zero after $K = 12$ under the CVT-based algorithm, but exhibits significant fluctuations under the VD-based algorithm even when K is large. As shown in (b) and (c), the CVT-based triangulation achieves a consistently lower deviation in cell size and edge length than the VD-based counterpart does.

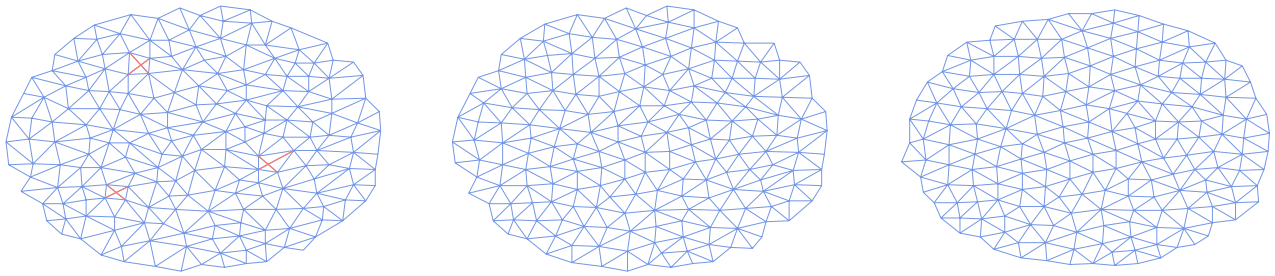


(a) VD-based triangulation for a C-shape sensor network. (b) CVT-based triangulation for a C-shape sensor network. (c) VD-based triangulation for a rectangle-shape sensor network. (d) CVT-based triangulation for a rectangle-shape sensor network.



(e) Localization result for VD-based triangulation. (f) Localization result for CVT-based triangulation. (g) Localization result for VD-based triangulation. (h) Localization result for CV-based triangulation.

Figure 9: Comparison of localization results with CVT-based and VD-based triangulations as inputs. A blue line segment is drawn for each node, starting from its real coordinates marked with blue dot and ending at the computed coordinates. The gray triangulation is based on the computed coordinates. The CVT-based triangulation supports more accurate localization than its VD-based counterpart does.



(a) Dual graph of Voronoi diagram (with crossing edges highlighted in red). (b) CVT-based triangulation based on cell boundary nodes. (c) CVT-based triangulation based on all nodes.

Figure 10: Similar triangulation result are achieved by the proposed CVT-based algorithm with either cell boundary nodes or all nodes (K=13).

is fundamentally unattainable to correctly judge neighboring cells when a Voronoi cell boundary is less than one hop. Consequently, the Voronoi diagram-based Delaunay triangulation fails.

This paper has proposed a distributed algorithm that performs centroidal Voronoi tessellation and constructs its dual graph to yield Delaunay triangulation. It exhibits several distinctive properties. First, it eliminates the problem due to short cell boundaries and thus effectively avoids crossing edges. Second, the proposed algorithm has been proven to converge and succeed in constructing a Delaunay triangulation, if the CVT cell size is greater than a constant threshold. Third, the established Delaunay triangulation consists of close-to-equilateral triangles, benefiting a range of applications such as geometric routing, localization, coverage, segmentation, and data storage and processing. Extensive simulations have been carried out under various 2D network models to evaluate the effectiveness and efficiency of the proposed CVT-based triangulation algorithm.

ACKNOWLEDGEMENT

M. Jin is partially supported by NSF CCF-1054996 and CNS-1018306. H. Wu is partially supported by NSF CNS-1018306, CNS-0831823, and CNS-0821702.

5. REFERENCES

[1] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. D. Gu, "Greedy routing with guaranteed delivery using ricci flows," in *Proc. of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 121–132, April 2009.

[2] R. Flury, S. V. Pemmaraju, and R. Wattenhofer, "Greedy Routing with Bounded Stretch," in *Proc. of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1737–1745, 2009.

[3] W. Zeng, R. Sarkar, F. Luo, X. D. Gu, and J. Gao, "Resilient Routing for Sensor Networks using Hyperbolic Embedding of Universal Covering Space," in *Proc. of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1–9.

[4] P. Bose and P. Morin, "Competitive Online Routing in Geometric Graphs," *Theoretical Computer Science*, vol. 324, no. 2, pp. 273–288, 2004.

[5] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based Localization of Large Scale Sensor Networks with Complex Shape," in *Proc. of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 789–797, 2008.

[6] M. Jin, G. Rong, H. Wu, L. Shuai, and X. Guo, "Optimal Surface Deployment Problem in Wireless Sensor Networks," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, pp. 2345–2353, 2012.

[7] M.-C. Zhao, J. Lei, M.-Y. Wu, Y. Liu, and W. Shu, "Surface Coverage in Wireless Sensor Networks," in *Proc. of the 28th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 109–117, 2009.

[8] H. Zhou, H. Wu, S. Xia, M. Jin, and N. Ding, "A Distributed Triangulation Algorithm for Wireless Sensor Networks on 2D and 3D Surface," in *Proc. of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1053–1061, 2011.

[9] R. Sarkar, W. Zeng, J. Gao, and X. Gu, "Covering space for in-network sensor data storage," in *Proc. of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 232–243, 2010.

[10] E. L. Lloyd, "On Triangulations of a Set of Points in the Plane," in *Proc. of the 18th Annual IEEE Symposium on Foundations of Computer Science*, pp. 228–240, 1977.

[11] H. Zhou, S. Xia, M. Jin, and H. Wu, "Localized algorithm for precise boundary detection in 3d wireless networks," in *Proc. in the 30th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 744–753, 2010.

[12] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational geometry: algorithms and applications*. Springer, 2008.

[13] S. Funke and N. Milosavljevi, "How Much Geometry Hides in Connectivity? - Part II," in *Proc. of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 958–967, 2007.

[14] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *Society for Industrial and Applied Mathematics (SIAM) review*, vol. 41, no. 4, pp. 637–676, 1999.

[15] D. Newman, "The hexagon theorem," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 137–139, 1982.

[16] D. Dong, Y. Liu, and X. Liao, "Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods," in *Proc. of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp. 135–144, 2009.

[17] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from Mere Connectivity," in *Proc. of ACM Int'l Symposium on Mobile Ad hoc Networking and Computing (MobiHOC)*, pp. 201–212, 2003.

[18] Y. Shang and W. Ruml, "Improved MDS-based Localization," in *Proc. of IEEE International Conference on Computer Communications*, pp. 2640–2651, 2004.

[19] G. Giorgetti, S. Gupta, and G. Manes, "Wireless Localization Using Self-Organizing Maps," in *Proc. of The International Symposium on ACM/IEEE International Conference on Information Processing in Sensor Networks (ACM/IEEE International Conference on Information Processing in Sensor Networks)*, pp. 293 – 302, 2007.

[20] L. Li and T. Kunz, "Localization Applying An Efficient Neural Network Mapping," in *Proc. of The 1st International Conference on Autonomic Computing and Communication Systems*, pp. 1–9, 2007.