Cut-and-Sew: A Distributed Autonomous Localization Algorithm for 3D Surface Wireless Sensor Networks

Yao Zhao
yxz4655@louisiana.edu

Hongyi Wu
wu@cacs.louisiana.edu

Miao Jin
mjin@cacs.louisiana.edu

Yang Yang
yxy6700@louisiana.edu

Hongyu Zhou
zhou.hongyu@me.com

Su Xia
suxia.ull@gmail.com

The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, USA

ABSTRACT

Location awareness is imperative for a variety of sensing applications and network operations. Although a diversity of GPS-less and GPS-free solutions have been developed recently for autonomous localization in wireless sensor networks, they primarily target at 2D planar or 3D volumetric settings. There exists unique and fundamental hardness to extend them to 3D surface. The contributions of this work are twofold. First, it proposes a theoretically-proven algorithm for the 3D surface localization problem. Seeing the challenges to localize general 3D surface networks and the solvability of the localization problem on single-value (SV) surface, this work proposes the *cut-and-sew* algorithm that takes a divide-and-conquer approach by partitioning a general 3D surface network into SV patches, which are localized individually and then merged into a unified coordinates system. The algorithm is optimized by discovering the minimum SV partition, an optimal partition that creates a minimum set of SV patches. Second, it develops practically-viable solutions for real-world sensor network settings where the inputs are often noisy. The proposed algorithm is implemented and evaluated via simulations and experiments in an indoor testbed. The results demonstrate that the proposed cut-and-sew algorithm achieves perfect 100% localization rate and the desired robustness against measurement errors.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

3D surface, autonomous localization, wireless sensor networks

1. INTRODUCTION

Location awareness is of significant importance to wireless sensor networks. It is imperative for a variety of sensing applications and network operations, ranging from position-aware sensing to sensor deployment and geometric routing. While location service can be readily provided by the global navigation system (GPS), it is often unaffordable to integrate a GPS receiver in every single sensor for large-scale deployment, due to its high cost and lavish energy consumption. Moreover, part or all of the sensors (e.g., deployed underground or underwater) may be prohibited from receiving line-of-sight satellite signals, rendering it infeasible to solely rely on GPS for sensor localization. To this end, a diversity of GPS-less and GPS-free solutions have been developed recently for *autonomous* localization in wireless sensor networks [1–26].

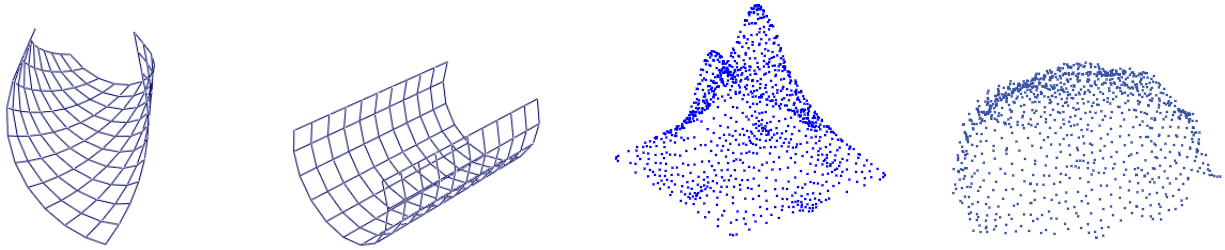
1.1 Challenges in 3D Surface Localization

A wireless sensor network may be deployed on a 2D plane (e.g., for crop sensing in fields or wildlife tracking on plains), or in a 3D volume (for underwater or space reconnaissance), or on a 3D surface (such as for seismic monitoring on ocean floors or in mountainous regions). Most autonomous localization algorithms are based on 2D sensor networks [1–24]. They take Euclidean distance information as input, and search the solution space to discover optimal sensor coordinates that minimize the average distance error (which is defined as the average difference between the real distance and the distance under the established coordinates system). The real Euclidean distance between two adjacent sensors can be approximately measured by received signal strength (RSS) or time difference of arrival (TDOA) or simply assumed as a constant radio range, while the real distance between two remote nodes is often estimated by their shortest path. A diversity of approaches have been proposed for distance error minimization in order to determine the coordinates of sensors, with different localization accuracy and time complexity [15–21]. Generally, distance information is sufficient to localize sensor nodes on a 2D plane (except for non-rigid shapes [27]). For example, Fig. 1(b) illustrates the localization result of the network shown in Fig. 1(a) by using multi-dimensional scaling (MDS) [15, 16].

It is straightforward to extend the 2D localization algorithms to 3D volume. Introducing the third dimension does not substantially increase the hardness of the problem. For instance, based on the estimated pair-wise Euclidean distances in a 3D volumetric network (shown in Fig. 1(c)), the MDS algorithm can be readily applied to establish the coordinates of the sensors (see Fig. 1(d)).

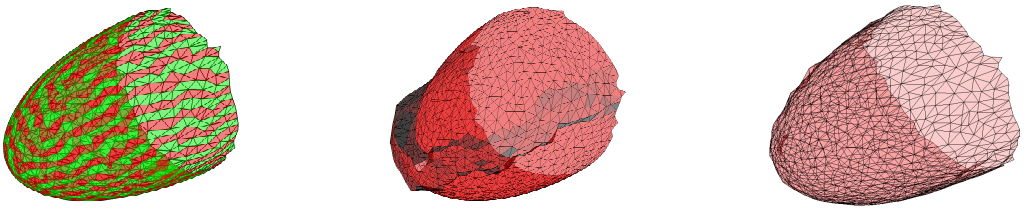
However, similar extension is not applicable to 3D surface networks as reported in [28]. While a 3D surface appears to be a special case of 3D volume or a generalization of 2D plane, surprising challenges exist in efforts to apply 2D planar or 3D volumetric lo-

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



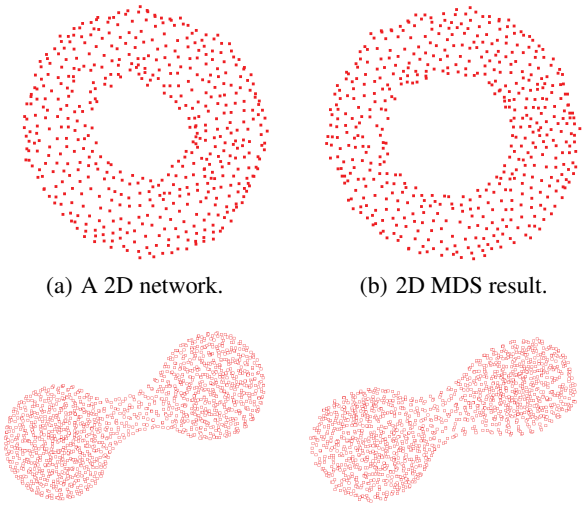
(a) A 3D surface. (b) Deformation of the surface. (c) A 3D surface network. (d) MDS result of (c).

Figure 2: A general 3D surface network is not localizable based on surface distances only, since it can be deformed to another surface without changing the surface distance between any pair of points as shown in (a) and (b). When a distance-based localization algorithm (e.g., MDS) is applied on 3D surface, it simply fails as illustrated in (c) and (d).



(a) Layer slicing [28]. (b) Localization result [28]. (c) Result of cut-and-sew.

Figure 3: The layer slicing approach proposed in [28] divides the surface into layers (see (a)). But it often fails in localizing some layers because they are NSV, thus resulting in large distortions in its localization result illustrated in (b). The proposed cut-and-sew algorithm produces much more accurate localization results as shown in (c).



(a) A 2D network. (b) 2D MDS result.

(c) A 3D volume network. (d) 3D MDS result.

Figure 1: Localization is feasible based on distance information in 2D plane and 3D volume sensor networks.

calization techniques to a 3D surface network. The hardness of the problem stems from the lack of Euclidean distance as input. More specifically, with short radio range, the measurable distance between two remote sensors on a 3D surface is their surface distance, which is essentially the length of geodesic, i.e., the (locally) shortest path between them on the surface. Such surface distance is often dramatically different from the corresponding 3D Euclidean distance. As revealed by Theorem 1 of [28], *a general 3D surface network is not localizable, given surface distance constraints only*. An intuitive explanation is depicted in Fig. 2, where the surface in Fig. 2(a) can be deformed to another surface (see Fig. 2(b) for example) without changing any surface distance. Thus, it is obviously impossible to determine a unique 3D embedding merely based on surface distances. When a distance-based localization algorithm (e.g., MDS) is applied to a 3D surface network, it simply fails as illustrated in Fig. 2(c)-2(d).

1.2 Contribution of This Work

Seeing the fundamental challenges in 3D surface localization based on surface distances only, a practical setting with augmented input information has been considered in [28], where not only surface distances but also nodal height measurements are assumed to formulate the localization problem. The height (or altitude) of a sensor is measurable via atmospheric pressure. Such measurement is of extremely low cost. As a matter of fact, many sensors have integrated barometer for gauging their altitude. For example, the Crossbow MTS400/MTS420 sensor board is equipped with Intersema MS55ER pressure sensor with an error margin of about 1.5% and thus able to determine its height with high accuracy. Similarly, the underwater height (or depth) may be measured via water pressure. Therefore, the height information can be taken along with surface distances as inputs to formulate the 3D surface localization problem.

At the first glimpse, the problem seems to become trivially easy with the given height (i.e., Z-coordinates) of sensors. For example, a naive approach is to project the sensors to X-Y plane and then apply 2D localization algorithms [15–17, 19–21] to determine their X-Y coordinates. The sensors are thus localized by putting X, Y and Z coordinates together. However this naive approach often fails because the projection of a general 3D surface on the X-Y plane is

non-planar. For instance, when the 3D surface network shown in Figs. 3 or 4 is projected to X-Y plane, the upper and lower parts of the surface will overlap, yielding a folded (i.e., non-planar) graph. The 2D localization algorithms either fail or result in extraordinarily large errors when they are applied to a significantly non-planar graph.

DEFINITION 1. *A 3D surface network is* localizable *if a unique embedding can be determined according to surface distance and height information.*

In general, a sensor network deployed on a single-value (SV) 3D surface is localizable. The formal definition of SV is to be given in Sec. 2.1. Briefly, a SV surface is a surface on which any two points have different projections on the X-Y plane. The definition is in reference to X-Y plane since sensors' heights are given as input. A more general definition of SV surface can be made according to any arbitrary plane, but is not considered in this research. Obviously, a network on a SV surface has a planar projection on X-Y plane, converting the problem to a 2D setting. Once the nodes are localized on 2D (i.e., X-Y coordinates are determined), they are mapped back to 3D by adding the height as Z-coordinate.

Seeing the challenges to localize general 3D surface networks and the solvability of the localization problem on SV surface, *this paper proposes a divide-and-conquer approach, dubbed cut-and-sew, by partitioning a general 3D surface network into SV patches, which are localized individually and then merged into a unified coordinates system.* The contributions of this work are twofold, including the design of theoretically-proven algorithm and the development of practically-viable solution.

(1) Theoretically-Proven Algorithm: There are obviously many options to partition a network. As a matter of fact, there is a theoretically infinite solution space to be explored. For example, a simple heuristic has been discussed in [28]. Given Z-coordinates of sensors, it is natural to divide a 3D surface network into short (e.g., one-hop high) horizontal layers, which are more likely to be SV and thus localizable. This approach does not guarantee all layers are SV. The non-single-value (NSV) layers are simply marked non-localizable. The localized layers are then combined together by least square alignment. Its localizable rate and location error highly depend on the percentage of NSV layers. Given a NSV surface, it is generally unavoidable to have NSV layers. Hence the localization result often exhibits significant distortion (as shown in Fig. 3(b)). Moreover, the algorithm may fail completely in a 3D surface network if most of its layers are NSV. This is just an example that an arbitrary partition does not yield the desired localization result.

This research aims to discover the minimum SV partition, an optimal partition that creates a set of patches satisfying two conditions. First, all patches must be SV to ensure their localizability. Second, the number of patches should be minimized to avoid unnecessary partitioning and merging, which are subject to linear transformation errors. This is obviously different from the trivially arbitrary partition adopted in several early works (such as [16,28]). The proposed approach is to identify NSV edges (as to be elaborated in Sec. 2.1) to guide the division of a 3D surface network into SV patches. Once the network is partitioned, the individual patches can be readily projected to 2D plane, where various algorithms are available for localization. Finally, the damped least-square algorithm [29] is employed to combine the localized patches by minimizing average distance error.

(2) Practically-Viable Solutions: Under practical sensor network settings, the inputs are often noisy. For example, both surface distances and sensors heights are subject to measurement errors. Although the basic ideas still apply, the noisy inputs obviously lead to inaccurate localization results or even a total failure of the localization algorithm. Practically-viable solutions must be developed to filter out input noise, aiming to improve the robustness and reliability and minimize localization errors. More specifically, the inaccurate distance and height measurement directly affects the identification of NSV edges, which are often deviated from the ground truth and become isolated. It is apparently impossible to partition the network according to such noisy NSV edges. The proposed idea is to fuse nearby NSV edges to form a band and then cut the network along the medial axis of the band. This approach effectively minimizes the impact of input errors on network partition and localization.

The propose algorithm is implemented and evaluated via simulations and experiments in an indoor testbed. The results demonstrate that the proposed cut-and-sew algorithm achieves nearly perfect 100% localization rate and the desired robustness against measurement errors. The rest of this paper is organized as follows: Sec. 2 introduces the proposed localization algorithm. Secs. 3 and 4 present testbed experiments and simulation results, respectively. Finally, Sec. 5 concludes the paper.

2. THE CUT-AND-SEW ALGORITHM

The inputs of the 3D surface localization problem include the height of the sensor and the connectivity and distance between neighboring nodes. The proposed distributed localization algorithm, named *cut-and-sew*, consists of three components as outlined below. First, it identifies NSV edges. Then, according to the NSV edges, the network is partitioned into a minimum set of SV patches that can be readily localized. Finally, the patches are merged together to produce a unified coordinates system. For a lucid exposition of the proposed scheme, the distance and height are first assumed free of errors. The problem due to measurement inaccuracy will be discussed in Sec. 2.4.

2.1 Identification of NSV Edges

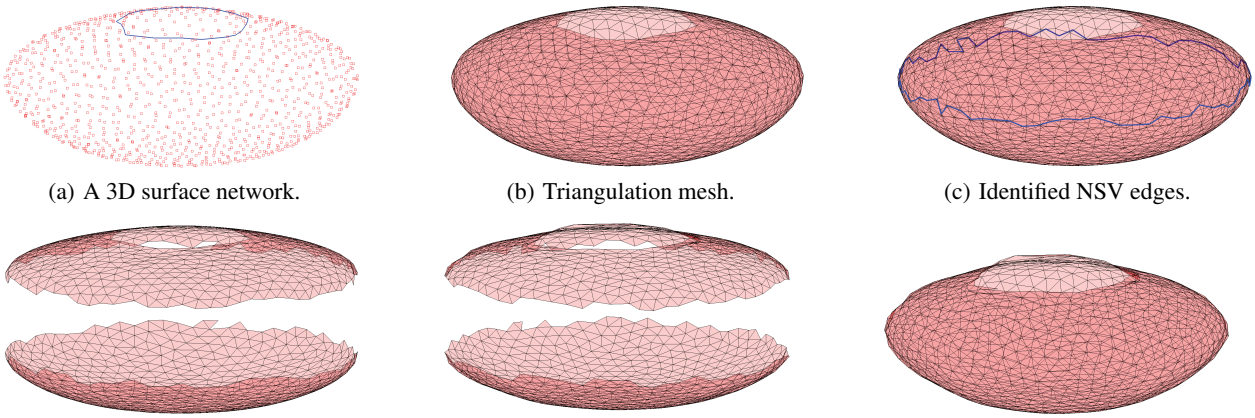
To facilitate network partitioning and localization, a distributed algorithm [30] is employed to establish a triangular mesh structure (or triangulation) based on local connectivity and distance information (see Fig. 4(a) and Fig. 4(b) for the original sensor network graph and the corresponding triangular mesh, respectively). If an edge in the triangular mesh is not on the boundary, it must be shared by two and only two triangles. For example, Fig. 5(a) illustrates two neighboring triangles, $\triangle ABC$ and $\triangle BCD$, which share a common edge, i.e., *BC*. Without loss of generality, the 3D surface network is oriented such that Edge *BC* is on the Y-Z plane. Let $\triangle abc$ and $\triangle bcd$ denote the projected triangles. Obviously, the projection of Edge *BC*, i.e., *bc*, is on the Y-axis. The length of each edge on the projected plane is determined according to the following equation:

$$L_{ij} = \sqrt{L_{IJ}^2 - (Z_I - Z_J)^2}, \tag{1}$$

where Z_I is the height of Node I, L_{IJ} is the length of Edge IJ, and L_{ij} is the length of Edge IJ's projection.

DEFINITION 2. *The local distance information of a node includes the Euclidean distances of the node to its one-hop and two-hop neighbors.*

The absolutely accurate distance measurement is generally unattainable in practice. However the Euclidean distance between two neighboring nodes can be estimated by RSS or TDOA. It often consumes higher power to measure the distance to a two-hop neighbor. But such measurement is required only once for a static network.



(a) A 3D surface network. (b) Triangulation mesh. (c) Identified NSV edges.

(d) Partitioned SV pieces. (e) Localization of individual pieces. (f) Final localization result.

Figure 4: An overview of the proposed cut-and-sew algorithm for 3D surface localization.

This is very different from using long links for communication, which results in significant power consumption.

DEFINITION 3. *In the triangulation of a 3D surface network, an edge is locally NSV (or NSV for short) if the projection of its two associated triangles overlap on the X-Y plane.*

It has been assumed in the above definition that neither triangles are vertical (i.e., the projection of a triangle is not colinear). The exception of vertical triangle will be discussed in Sec. 2.3. It can be checked by a simple, local test and in fact leads to a trivial problem that is readily solvable.

DEFINITION 4. *A 3D surface sensor network is called a NSV network if it contains NSV edges.*

DEFINITION 5. *A 3D surface sensor network is called a SV network if it does not contain NSV edges.*

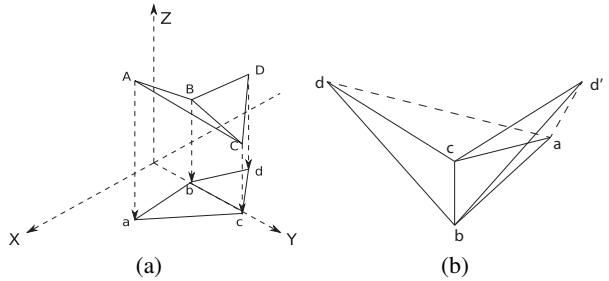
As discussed earlier, the NSV network, particularly the NSV edges, introduce problem in localization. The proposed algorithm, however, exploits them to partition the network into SV patches that can be readily localized. The rest of this subsection shows that NSV edges can be identified by using local information.

LEMMA 1. *Given an edge in the triangular mesh of a 3D surface sensor network, its associated local distance information is sufficient to determine whether it is a NSV edge.*

PROOF. Consider Edge BC shared by $\triangle ABC$ and $\triangle BCD$, which are projected to $\triangle abc$ and $\triangle bcd$ on the 2D plane. Obviously, to check if $\triangle abc$ and $\triangle bcd$ overlap is equivalent to examine whether Nodes a and d are on the same side of Edge bc. If Nodes a and d are on the same side of Edge bc, $\triangle abc$ and $\triangle bcd$ overlap; otherwise, they do not.

Without loss of generality, Node a is assumed on an arbitrary side of Edge bc. Now, the problem is reduced to check if Node d is on the same side. Assume the coordinates of Nodes b and c are (X_b, Y_b) and (X_c, Y_c). Given the edge length information, i.e., L_{bd} and L_{cd}, the basic triangulation can be applied to derive the coordinates of Node d. Clearly, there are two possible solutions symmetric about Y-axis, denoted as d' and d'' with coordinates (X_d, Y_d) and $(-X_d, Y_d)$, respectively, as illustrated in Fig. 5(b).

So far it still unknown on which side Node d is located. However, it can be shown by contradiction that the distances between



(a) (b)

Figure 5: (a) Projections of neighboring triangles. (b) Hypothetic nodal positions become deterministic with 2-hop distance.

Node a and the two hypothetic positions of Node d (i.e., d' and d'') are always different. If $L_{ad'} = L_{ad''}$, then

$$(X_a - X_d)^2 + (Y_a - Y_d)^2 = [X_a - (-X_d)]^2 + (Y_a - Y_d)^2. \quad (2)$$

The solution of the equation is $X_a = 0$ or $X_d = 0$, i.e., either Node a or Node d must be collinear with Nodes b and c. This contradicts the fact that Nodes a, b, c and d form two triangles. Therefore, if L_{ad} is known, one can readily determine whether Node a and Node d are on the same side of Edge bc. More specifically, one can first embed $\triangle abc$ on the 2D plane according to the local distances L_{ab}, L_{bc}, and L_{ac}. Then, based L_{bd} and L_{cd}, two hypothetic positions of Node d are computed. Finally, one of them is chosen according to L_{ad}. If Nodes a and d are on the same side, Edge BC is NSV; otherwise, it is not. Thus the lemma is proven. □

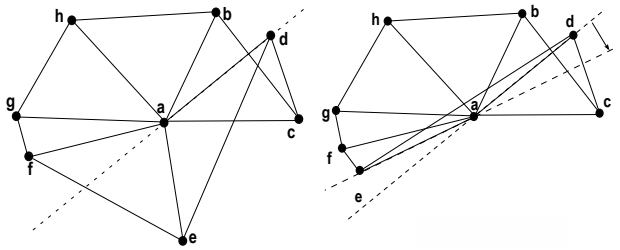
The proof of Lemma 1 clearly suggests a simple and localized scheme to examine if an edge is NSV. Every node in the network can perform such local calculation for its associated edges and mark the ones that are NSV. Fig. 4(c) illustrates the identified NSV edges.

2.2 Network Partition

This subsection shows that the minimum SV partition is achieved by dividing the network along NSV edges.

DEFINITION 6. *Given a triangular mesh, a boundary edge is an edge contained in one and only one triangle.*

A boundary edge is obviously not NSV because it is contained in one triangle only.



(a) Node e on right side of ad. (b) Node e on left side of ad.

Figure 6: Illustration of Lemma 2, where Edge ac is NSV.

LEMMA 2. *A NSV edge must connect to other NSV edges or boundary edges.*

PROOF. Consider a NSV edge, denoted as Edge AC. Its projection on the 2D plane is ac. First, Edge AC must be shared by two triangles according to Definition 3. Therefore it is not a boundary edge. If either Node A or Node C is on boundary, Edge AC must connect to a boundary edge, and thus the lemma is proven.

Now assume Nodes A and C are non-boundary nodes. A non-boundary node must be surrounded by a set of triangles on the 3D surface. Fig. 6 illustrates the projection of the set of triangles around Node A on the 2D plane. Since Edge AC is NSV, the projection of its associated triangles must overlap. In other words, Nodes b and d must be on the same side of ac on the projected plane. Thus either ad is located within $\angle bac$ or ab is in $\angle dac$. Since the two cases are symmetric, only the first one is discussed here. Given ad is within $\angle bac$, Nodes b and c are obviously on the two sides of Line ad (see the dashed line in Fig. 6). To facilitate the discussion, a directions is defined as follows. Assume one stand at Node a and face Node d. Node c is on the right hand side, thus it is said on the right side of ad. Similarly, Node b is on the left side of ad. Besides left and right, the following discussion also uses the clockwise order around Node a.

Next the proof shows by contradiction that besides Edge AC, there must exists another NSV edge associated with Node A. To construct the contradiction, all edges that connect to Node A, except AC, are assumed SV. Consequently, the neighboring nodes of Node A (excluding C) must be in a clockwise order on the projected plane. More specifically, let's begin with Node d as shown in Fig. 6(a). Without loss of generality, assume d is on the left side of ae. Since ae is SV, Node f must be on its right side. Thus Nodes d, e and f must be in a clockwise order around Node a. Similarly, since af is SV, Nodes e, f, and g must follow the clockwise order. Thus Nodes d to g are all ordered. By deduction, all nodes around Node a, except c, must be clockwise ordered (see Nodes d, e, ..., b in the figure).

Now the proof shows that either Edge AD or Edge AE must be NSV. First, examine $\triangle dae$, which shares ad with $\triangle cad$. As discussed earlier, Node c is on the right side of ad. If Node e is also on the right side of ad (as illustrated in Fig. 6(a)), then Edge AD must be NSV. If Node e is on the left side of ad (as illustrated in Fig. 6(b)), let the dashed line ad rotate clockwise around Node a. Since Nodes d, e, ..., b are clockwise ordered, the dashed line must meet Node e first. Obviously, both d and f are on the same side of ae. Accordingly, Edge AE is NSV. The above results contradict the earlier assumption that all edges that connect to Node A, except AC, are SV. Thus besides Edge AC, there must exists another NSV edge associated with Node A.

The above discussions focuses on Node A and its surrounding triangles. Similar results can be obtained by analyzing Node C and its neighboring edges. Therefore, the proof concludes that the two ends of an NSV edge must connect to other NSV edges or boundary edges. The lemma is proven. □

In fact, the above lemma is intuitively understandable, because the NSV edge is where the surface is folded when it is projected to the X-Y plane. Given a 3D surface, the folding line must be continuous until it extends to the boundary of the surface or meets other folding lines.

THEOREM 1. *The minimum SV partition is achieved by dividing the network along NSV edges.*

PROOF. Let the partition process start from any node on an arbitrary NSV edge and cut the network along all of its connected NSV edges. Since the NSV edges must connected to each other or to boundary edges as shown by Lemma 2, the cutting process will either form a loop or stop at the boundaries of the network. In either case, the network is partitioned into two or more separated patches. The NSV edges used in such partition become boundary edges of the newly created patches. As a result, they are no longer NSV edges, because a boundary edge is contained in one triangle only. The process repeats until no NSV edges exist in the entire network. It is clear that none of the patches contains an NSV edge. Therefore they are all SV.

On the other hand, it is straightforward to show the partition is minimum, because all NSV edges must be cut open, otherwise a patch that contains NSV edges must be an NSV patch. The theorem is thus proven. □

Fig. 4(d) illustrates the SV patches by partitioning the 3D surface sensor network along NSV edges.

2.3 Localization and Combination

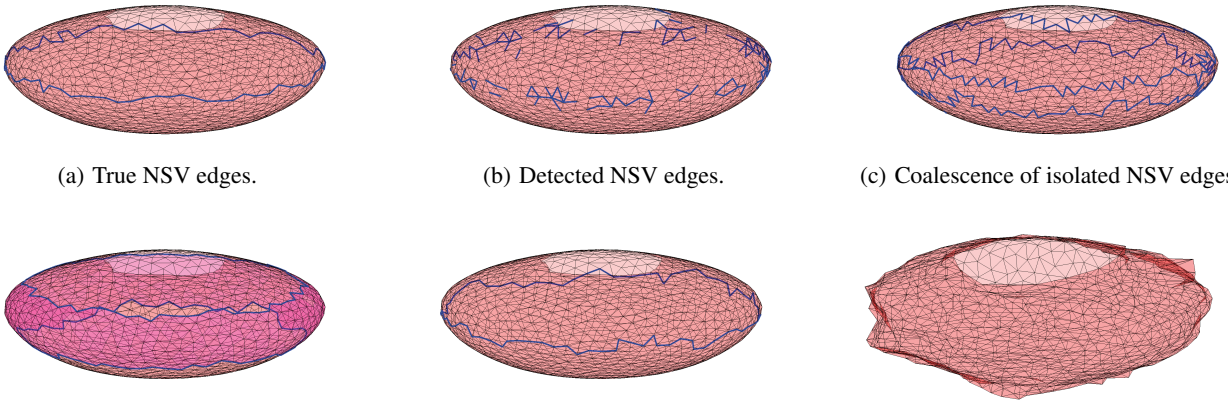
As discussed above, each edge decides if it is NSV by using local information only. A set of NSV edges form the boundary of an SV patch. At least one randomly elected node in each patch initiates projection and 2D localization by constructing a local flooding packet that contains a patch ID (which, e.g., can be simply its own node ID). The packet is dropped when it reaches the NSV edges. Obviously the local flooding packet is limited within the given patch. If multiple nodes flood at the same time, the one with the highest ID wins. All nodes in the patch thus begin projection and 2D localization.

The projection of a SV patch is planar. It remains a triangulation on the X-Y plane, with only changes in edge length that can be determined by surface edge length and height information (i.e., by using Eq. (1)). Given the height information, Z-coordinates are already known. Thus only X-Y-coordinates are yet to be determined. To this end, several distributed 2D localization algorithms can be applied [15–17, 31]. The localization result is then mapped back to 3D by including the known Z-coordinates (see Fig. 4(e)).

There is a rare exceptions of Definition 3 that may occur when a triangle is vertical. A vertical triangle can be checked by a simple, local test since it is colinear on the X-Y plane. If one and only one triangle associated with an edge is vertical, the edge is considered a NSV edge. However, if both triangles are vertical, the edge is not treated as NSV.

Similarly, an entire patch may be vertical. Its projection becomes a line, which is not localizable by 2D algorithms. However, the patch is in fact already planar (without a projection), and thus can be readily localized on the X-Y plane and mapped to 3D as discussed above.

Finally, since neighboring patches share common vertexes and edges, they can be readily "sewed" together by using a distributed least square algorithm [28,32], yielding a unified coordinate system for the entire 3D surface sensor network as illustrated in Fig. 4(f).



(a) True NSV edges. (b) Detected NSV edges. (c) Coalescence of isolated NSV edges.

(d) Formation of NSV band. (e) Partition along medial axis. (f) Final localization result.

Figure 7: Network partition under noisy measurements.

2.4 Practical Solution with Noisy Inputs

So far accurate distance and height measurements are assumed as inputs of the 3D surface localization problem. Both of them can be noisy under practical sensor network settings. The proposed algorithm still applies but needs further treatment on network partitioning.

The inaccurate inputs directly affect the identification of NSV edges. Fig. 7(b) illustrates an example of the detected NSV edges with 10% measurement errors. As can be seen, they become isolated and many of them are deviated from the true NSV edges shown in Fig. 7(a). It is obviously impossible to partition the network according to such noisy NSV edges directly. The proposed idea is to fuse the nearby NSV edges to form a band and then cut the network along the medial axis of the band. More specifically, the algorithm consists of the following three steps.

(1) Coalescence of Isolated NSV Edges. The algorithm fuses the detected NSV edges by expanding and connecting them. First, if a triangle contains an NSV edge, the entire triangle, i.e., all of its edges, are marked NSV. Second, If two NSV triangles are one-hop away from each other, the edges between them are marked as NSV too. The NSV edges are now better connected, forming a number of clusters. The NSV edges in each cluster are connected, but different clusters are still isolated. The two closest clusters are connected by marking all the edges on their shortest path as NSV edges. They are thus merged into one cluster. The process repeats until all clusters are connected. Fig. 7(c) shows the result after coalescence of isolated NSV edges.

(2) Formation of NSV Band. The above step has created an expanded set of NSV edges. The algorithm further marks the edges within 1-hop of existing NSV edges as NSV, forming a NSV band. To smoothen the band, an edge is marked NSV if it is included in a triangle with two NSV edges. An example of the resulting NSV band is depicted in Fig. 7(d).

(3) Partition along Medial Axises of NSV Band. Finally, the medial axis of the NSV band is identified for partitioning. A NSV band may have two or more boundaries. The basic idea is to shrink the boundaries of the NSV band until they meet, forming the approximate medial axis. This is achieved by a distributed process initiated by boundary edges of the NSV band. Each boundary edge is involved in a triangle inside the band. It is replaced by two other edges of the triangle, which become new boundary edges. Thus the boundaries grow inward into the band. The process repeats until convergence (which is obviously guaranteed because the band has a finite size), yielding the medial axis.

Once the network is partitioned, the patches are localized and combined as discussed earlier. Note that the partition along medial axis is an approximation. It does not guarantee every patch is SV. Thus, the 2D projection of a patch may have minor overlap between its edges (especially at the boarder of the patch), resulting in localization errors. This is evident from the minor distortion shown in Fig 7(f). Such errors will be quantitatively studied and discussed in Sec. 4.

2.5 Complexity and Overhead

The NSV edge identification, network partitioning and projection are all done locally by the individual nodes, thus resulting in a constant computation complexity and communication overhead. However, to initiate projection, a local flooding is required in all individual patches, which together yield a communication overhead of $O(n)$ in the entire network, where n is the total number of nodes in the network.

When inputs are noisy, the NSV edges are further processed to identify the medial axis of the NSV band. This process consumes a total time of $O(n)$ and the corresponding communication overhead is also $O(n)$.

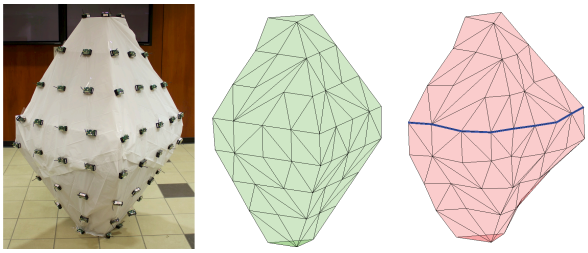
The computation time and communication overhead for localizing individual patches depend on the 2D localization algorithm employed. For example, MDS results in a linear communication overhead and a computation complexity of $O(m^3)$ where m is the number of nodes in a patch. The computation is carried out in all patches simultaneously.

The combination of two patches can be completed in $O(p)$ time and results in $O(p)$ communication overhead, where p is the set of nodes involved in alignment (usually the common nodes shared by the two patches). To merge all patches, the total computational complexity and communication overhead are both $O(n)$.

In summary, the proposed cut-and-sew algorithm results in an overall communication overhead of $O(n)$ and time complexity of $O(Max\{m^3, n\})$.

3. PROTOTYPING AND EXPERIMENTS

Several indoor testbed models have been built in this research for prototyping 3D surface sensor networks. An example is shown in



(a) (b) (c)

Figure 8: Experimental setup and result. (a) Indoor 3D surface network testbed. (b) Input triangulation. (c) Localization result (with an average location error of 14%).

Table 1: Localizable Rate

	Stadium	Sea Cave	Mine Pit
Slice-Based [28]	98.8%	90.8%	81.3%
Cut-and-Sew	100%	100%	100%

Fig. 8(a), which is $5^1/_4$ feet high, 3 feet long and 3 feet wide. Forty eight Crossbow MICAz motes are attached to its surface. The algorithmic codes are implemented in Tiny OS and run on the Crossbow sensor nodes. The sensors are configured to use close to minimum radio transmission power (Level 2), with a communication range between 25 to 55 cm. The short radio range avoids undesired connections through volume.

Every sensor periodically broadcasts a beacon message that contains its node ID to its neighbors. Based on received beacon messages, a node builds a neighbor list with the RSSI of corresponding links. RSSI is used to estimate the length of links by looking up a RSSI-distance table established by experimental training data. The preliminary test shows that, under low transmission power, such estimation has an error rate about 20%. At the same time, the ground truth of surface distances and sensor coordinates are manually measured. A triangular mesh structure is constructed by using the distributed algorithm [30]. Fig. 8(b) illustrates the triangulation based on ground truth inputs.

The localization result is depicted in Fig. 8(c). As can be seen, the NSV edges are identified correctly (as highlighted in the figure). The network is thus partitioned into two SV patches. Then MDS is applied to localize each of them. The combined patches largely restore the original 3D surface network, with an average location error of about 14%. The largest errors are observed at the middle of the network (around the NSV edges), while the top and bottom are localized more accurately. This is because the triangles right below the NSV edges are almost vertical, resulting in extremely skinny projections on the X-Y plane (where some edges of the projected triangles are extremely short). Note that localization errors are inevitable in MDS, due to inaccurate distance inputs. Given the same amount of errors, they apparently have larger impact on short edges, which lead to significant distortion on 3D surface, thus producing the aforementioned localization errors.

The layer slicing approach proposed in [28] largely fails in the above experimental setting, because it cannot locate about 50% of sensor nodes (around the NSV edges at the middle of the network).

4. SIMULATION RESULTS

Besides the indoor testbed experiments, the proposed cut-and-sew algorithm is implemented and evaluated via simulations under a variety of practical 3D surface sensor networks. For example, the results under three representative network models (for monitoring stadium, sea cave, and mine pit) are presented in this section. Note that the localization results obviously depend on individual network settings. It would mix up the performance trend or even become misleading if data from different networks are combined and averaged.

The simulation adopts a general communication model with merely a constraint on the maximum radio transmission range, which is normalized to one. Two nodes are connected with a probability if their distance is less than one. Each node is assumed to measure its own height and local distances. A range of measurement errors are considered in simulations. The proposed algorithm has been proven to work perfectly when the measurements are accurate. Therefore, the simulations focus on studying its tolerance against measurement errors and the impact of sensor densities. There is almost no competing schemes to compare with. The only related work is [28]. However, it is difficult to compare their localization errors, because the approach in [28] does no even localize all nodes in the network. Therefore, the comparison is based on localizable rate only.

4.1 NSV Edge Detection Error

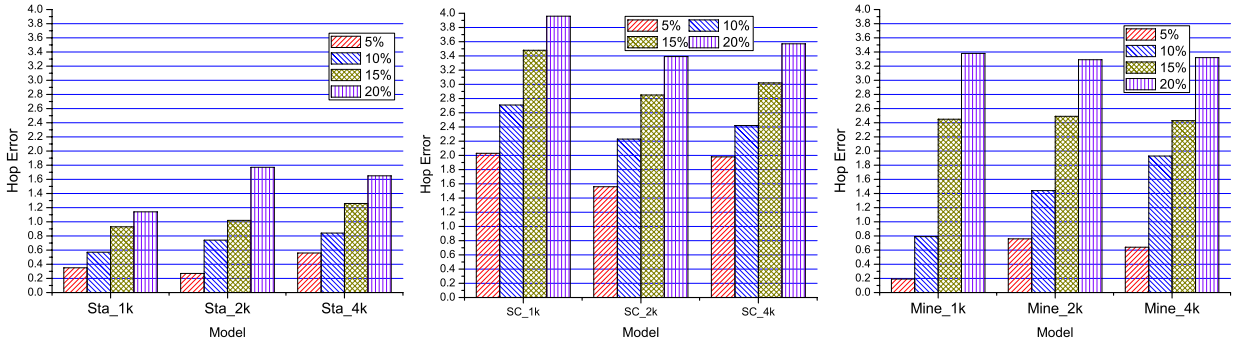
The proposed algorithm heavily depends on NSV edges, which guide the partition of the network. Let S denote the set of perfect NSV edges (identified under precise distance and height measurements). While the measurements are often noisy in practice, the proposed algorithm still yields a set of NSV edges, T. Obviously, the accuracy of network partition and final localization depend on how close T is to S. Hence, the NSV edge detection error is defined as $E_{NSV} = \frac{\Sigma_{t_k \in T} N(t_k,S)}{|T|}$, where $N(t_k,S)$ is the hop distance between Edge t_k and its closest edge in S, and $|T|$ is the total number of identified NSV edges.

Fig. 9 illustrates the impact of measurement errors and nodal densities on E_{NSV}. Under the same nodal density, E_{NSV} increases sharply with larger measurement errors. While the results show undesired vulnerability of NSV edge detection under measurement errors, the simulation data to be discussed next demonstrates that the proposed algorithm effectively minimizes the impact of the NSV edge errors on network partition and final localization. On the other hand, the nodal density (varying from 1k to 4k in each network model) does not noticeably affect NSV edge detection.

4.2 Network Partition Error

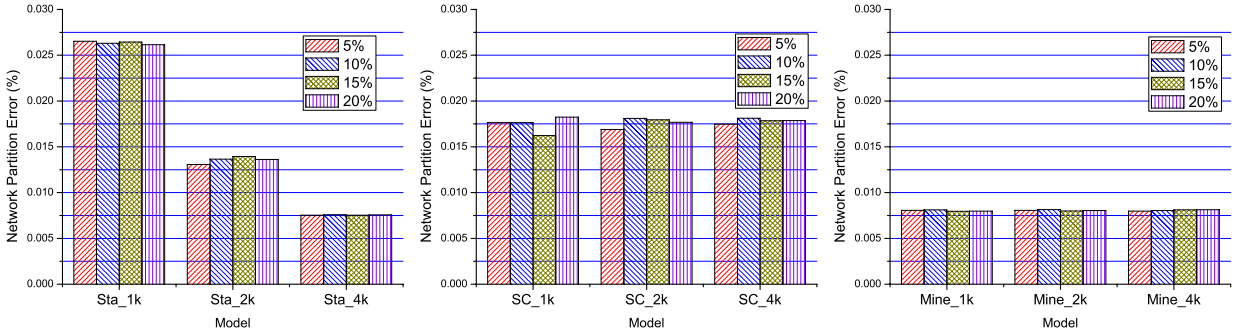
The network partition is guided by the NSV edges. Therefore, NSV edge errors obviously have a negative impact on partitioning. However, the proposed algorithm employs several techniques to minify the effect. More specifically, it fuses the nearby NSV edges to form a band and then identifies the medial axis of the band for network partitioning. The medial axis is expected to approximate the set of true NSV edges. Let U denote the set of edges that constitutes the identified medial axis. The network partition error E_{PAR} is defined as the maximum deviation between U and S (i.e., the set of true NSV edges for ideal partitioning), i.e., $E_{PAR} = \frac{\max\{N(u_i,S)|u_i \in U\}}{M}$, where $N(u_i,S)$ is the shortest hop distance between u_i and S, and M is a normalizer defined as the maximum shortest hop distance between any two nodes in the network. Higher E_{PAR} indicates that, after the network is partitioned, more NSV areas still exist in the patches, leading to potentially worse localization result.

As shown in Fig. 10, E_{PAR} is not sensitive to measurement errors. With the increase of measurement inaccuracy, the network partition error largely remains a constant. Nodal density has a diverse effect



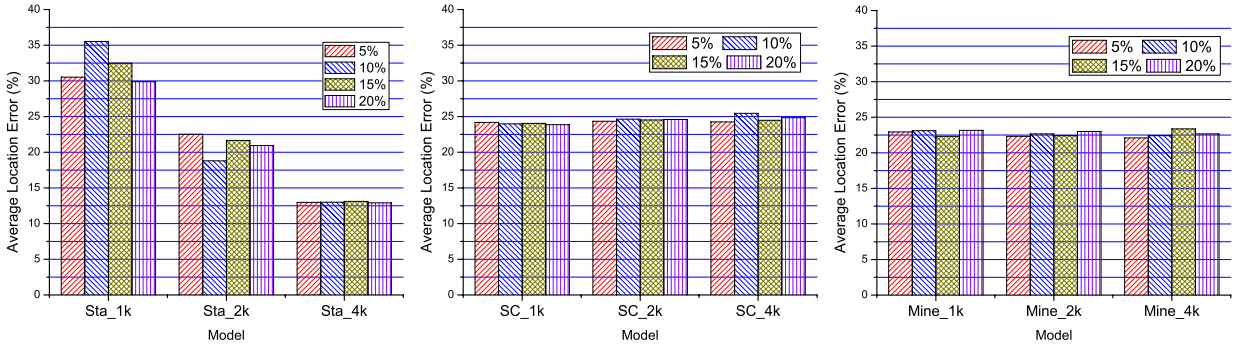
(a) Stadium model. (b) Sea cave model. (c) Mine pit model.

Figure 9: NSV edge detection error increases dramatically with higher measurement errors. Each model is simulated with 1k, 2k and 4k nodes.



(a) Stadium model. (b) Sea cave model. (c) Mine pit model.

Figure 10: Network partition error is insensitive to measurement errors. Nodal density has a diverse effect on network partition error depending on individual network models.



(a) Stadium model. (b) Sea cave model. (c) Mine pit model.

Figure 11: Average localization error is not significantly affected by measurement errors.

on network partition error depending on individual network models. For example, the impact of nodal density is negligible under the sea cave and mine pit models. However, the stadium model noticeably benefits from higher nodal density, owing to the shape of the area surrounding NSV edges.

4.3 Average Location Error

The final output of the algorithm is the coordinates of sensor nodes. Such coordinates can be aligned to any desired coordinates system (e.g., the global positioning system) by using the least square algorithm [32]. However, the alignment to a particular coordinates system is out of the interests of this research. Therefore, the localization error is computed by comparing the edge length under the established coordinates system with the real edge length. More specifically, let $\{l_1, l_2, ..., l_n\}$ denote the edge lengths calculated according to the established coordinates and $\{l'_1, l'_2, ...l'_n\}$ the real edge lengths. Location error E_{LOC} is defined as $E_{LOC} = \frac{\sum_{k=1}^{n} |l_k - l'_k|}{n \times \max\{l'_i | 1 \le i \le n\}}$.

It is not a surprise that the localization error shows similar trend as the network partition error. As a matter of fact, the performance of localization is predominated by the errors in network partitioning. With an average localization error less than 25%, the localization algorithm is not significantly affected by inaccurate distance and height measurements (see Fig. 11). A higher sensor density helps reduce the localization errors in the stadium model. However, the same effect is not observed in the sea cave and mine pit models, in line with the tread in Fig. 10.

Finally, the proposed algorithm achieves 100% localizable rate, which is in a sharp contrast to the earlier heuristic [28]. Table 1 shows a comparison of the localizable rate (averaged over different sensor densities).

5. CONCLUSION

This paper has proposed a divide-and-conquer approach, named cut-and-sew, for autonomous localization of 3D surface wireless sensor networks. Seeing the challenges to localize general 3D surface networks and the solvability of the localization problem on single-value (SV) surface, the proposed cut-and-sew algorithm partitions a general 3D surface network into SV patches, which are localized individually and then merged into a unified coordinates system. The algorithm has been optimized by discovering the minimum SV partition, an optimal partition that creates a minimum set of SV patches. Moreover, the paper has introduced a practically-viable solution for real-world sensor network settings where the inputs are noisy. The proposed algorithm has been implemented and evaluated via simulations and indoor testbed experiments. The results have demonstrated perfect 100% localization rate and the desired robustness against measurement errors.

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.

6. REFERENCES

[1] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, 2000.

[2] A. Savvides, C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proc. of MobiCom*, pp. 166–179, 2001.

[3] J. Albowicz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks," in *Proc. of ICNP*, pp. 35–41, 2001.

[4] L. Doherty, L. Ghaoui, and K. Pister, "Convex Position Estimation in Wireless Sensor Networks," in *Proc. of INFOCOM*, pp. 1655–1663, 2001.

[5] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free Localization Schemes in Large Scale Sensor Networks," in *Proc. of MobiCom*, pp. 81–95, 2003.

[6] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," in *Proc. of GLOBECOM*, pp. 2926–2931, 2001.

[7] C. Savarese and J. Rabaey, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *Proc. of USENIX Annual Technical Conference*, pp. 317–327, 2002.

[8] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 105–111, 2002.

[9] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *Proc. of INFOCOM*, pp. 1734–1743, 2003.

[10] H. S. AbdelSalam and S. Olariu, "Passive Localization Using Rotating Anchor Pairs in Wireless Sensor Networks," in *Proc. of The 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking*, pp. 67–76, 2009.

[11] Z. Zhong and T. He, "MSP: Multi-Sequence Positioning of Wireless Sensor Nodes," in *Proc. of SenSys*, pp. 15–28, 2007.

[12] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A Theory of Network Localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[13] K. Yedavalli and B. Krishnamachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 81–94, 2008.

[14] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-Free Positioning in Mobile Ad-Hoc Networks," in *Proc. of The 34th Annual Hawaii International Conference on System Sciences*, pp. 3481–3490, 2001.

[15] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from Mere Connectivity," in *Proc. of MobiHoc*, pp. 201–212, 2003.

[16] Y. Shang and W. Ruml, "Improved MDS-based Localization," in *Proc. of INFOCOM*, pp. 2640–2651, 2004.

[17] V. Vivekanandan and V. W. S. Wong, "Ordinal MDS-based Localization for Wireless Sensor Networks," *International Journal of Sensor Networks*, vol. 1, no. 3/4, pp. 169–178, 2006.

[18] H. Wu, C. Wang, and N.-F. Tzeng, "Novel Self-Configurable Positioning Technique for Multi-hop Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 609–621, 2005.

[19] G. Giorgetti, S. Gupta, and G. Manes, "Wireless Localization Using Self-Organizing Maps," in *Proc. of IPSN*, 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.

[21] M. Jin, S. Xia, H. Wu, and X. Gu, "Scalable and Fully Distributed Localization With Mere Connectivity," in *Proc. of INFOCOM*, pp. 3164–3172, 2011.

[22] J. Wang, M. Tian, T. Zhao, and W. Yan, "A GPS-Free Wireless Mesh Network Localization Approach," in *Proc. of International Conference on Communications and Mobile Computing*, pp. 444–453, 2009.

[23] R. Magnani and K. K. Leung, "Self-Organized, Scalable GPS-Free Localization of Wireless Sensors," in *Proc. of WCNC*, pp. 3798–3803, 2008.

[24] H. Akcan, V. Kriakov, H. Bronnimann, and A. Delis, "GPS-Free Node Localization in Mobile Wireless Sensor Networks," in *Proc. of The 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 35–42, 2006.

[25] C. Wang, H. Wu, and N.-F. Tzeng, "RFID-Based 3-D Positioning Schemes," in *Proc. of INFOCOM*, pp. 1235–1243, 2007.

[26] J. Maneesilp, C. Wang, H. Wu, and N. F. Tzeng, "RFID Support for Accurate 3-Dimensional Localization," *IEEE Transactions on Computers*, vol. 62, no. 7, pp. 1447–1459, 2013.

[27] Z. Yang, Y. Liu, and X.-Y. Li, "Beyond Trilateration: On the Localizability of Wireless Ad-Hoc Networks," in *Proc. of INFOCOM*, pp. 2392–2400, 2009.

[28] Y. Zhao, H. Wu, M. Jin, and S. Xia, "Localization in 3D surface sensor networks: Challenges and solutions," in *Proc. of INFOCOM*, pp. 55–63, 2012.

[29] K. Levenberg, "A Method for the Solution of Certain Non-linear Problems in Least-Squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[30] 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 INFOCOM*, pp. 1053–1061, 2011.

[31] H. Lim and J. Hou, "Distributed Localization for Anisotropic Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, pp. 11:1–11:26, 2009.

[32] A. Bjorck, *Numerical Methods for Least Squares Problems*. No. 51, Society for Industrial Mathematics, 1996.