CONSEL: Connectivity-based Segmentation in Large-Scale 2D/3D Sensor Networks

Hongbo Jiang[1] Tianlong Yu[1] Chen Tian[1] Guang Tan[2] Chonggang Wang[3]

[1]Department of Electronics and Information Engineering, Huazhong University of Science and Technology, China
[2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China,
[3]InterDigital Communications, U.S.A.
[1]{hongbojiang2004,yutianlong21,alexandretian}@gmail.com,[2]guang.tan@siat.ac.cn, [3]cgwang@ieee.org

***Abstract*—A cardinal prerequisite for the system design of a sensor network, is to understand the geometric environment where sensor nodes are deployed. The global topology of a large-scale sensor network is often complex and irregular, possibly containing obstacles/holes. A convex network partition, so-called segmentation, is to divide a network into convex regions, such that traditional algorithms designed for a simple geometric region can be applied. Existing segmentation algorithms highly depend on concave node detection on the boundary or sink extraction on the medial axis, thus leading to quite sensitive performance to the boundary noise. More severely, since they exploit the network's 2D geometric properties, either explicitly or implicitly, so far there has been no general 3D segmentation solution.**

In this paper, we bring a new view to segmentation from a Morse function perspective, bridging the convex regions and the Reeb graph of a network. Accordingly, we propose a novel distributed and scalable algorithm, named CONSEL, for CONnectivity-based SEgmentation in Large-scale 2D/3D sensor networks. Specifically, several boundary nodes first perform flooding to construct the Reeb graph. The ordinary nodes then compute mutex pairs locally, thereby generating the coarse segmentation. Next the neighbor regions which are not mutex pair are merged together. Finally, by ignoring mutex pairs which leads to small concavity, we provide the constraints for approximately convex decomposition. CONSEL is more desirable compared with previous studies: (1) it works for both 2D and 3D sensor networks; (2) it only relies on network connectivity information; (3) it guarantees a bound for the regions' deviation from convexity. Extensive simulations show that CONSEL works well in the presence of holes and shape variation, always yielding appropriate segmentation results.

I. Introduction

Recent years wireless sensor networks have been witnessed a wide usage for many critical applications, such as habitat monitoring and battlefield surveillance. A cardinal prerequisite for efficient design of a sensor network is to understand the geometry of the environment where sensor nodes are deployed. The global topology of a large-scale sensor network is often complex and irregular, possibly containing obstacles/holes. As such, a sensor field is rarely in a simple shape such as a square or a disk, as assumed in many research studies (e.g., [4], [18]). For example, geographic routing [10] is a typical routing scheme for wireless sensor networks, where a node makes routing decisions greedily, often based on a small set of local coordinates. Specifically, a node routes a message to its neighbor closest to the destination. Despite its success on a regular sensor field, this protocol performs poorly in irregular and concave areas [11], [12], [19], [22]. Also, irregular network shapes may lead to inaccurate localization results, as many existing localization algorithms assume straight-line shortest paths between nodes. This does not hold when the network topology is highly irregular, resulting in an over-estimated shortest path length [13], [20] and thus inaccurate localization results.

Numerous methods [11]–[13] have been proposed to improve the traditional algorithms' performance by adapting them to irregular sensor field. On the downside, these efforts are mostly application-specific, thereby increasing the complexity significantly. Alternatively, to tame the challenges brought by irregular shape, there has been an increasing interest on a convex network partition, so-called *segmentation/convex partition* [19], [22] (in this paper, we consider shape segmentation instead of data segmentation [17] or signal field segmentation [23]), which is to divide a network into convex regions/subregions/subnetworks, such that traditional algorithms designed for a simple geometric region can be applied. By doing so, without heavily modifying particular algorithms, traditional algorithms are able to perform well in each convex region, no need to consider a deviation of many geometric elements such as hop distance along a straight-line shortest path.

A. Related Work

A pioneer piece of work on sensor network segmentation is [22]. It first extracts the media axis and constructs the distance field of a network. Next, based on flow complex [2], a node on a flow is assigned a flow direction and identifies itself a sink if there is no flow direction. According to the flow direction to the sink, the ordinary nodes are grouped to individual regions. Inspired by polygonal partition, the solution in [19] partitions the network via concave node identification on the boundary. The main idea of their algorithm is to perform bisector-induced convex partitioning. These segmentation algorithms, unfortunately, depend on the existence of nodes on distinguished concave boundaries, and thus they may fail for the networks where no such nodes (or sinks on the media axis) exist, as shown in Fig. 1. The reason is that their algorithms do not consider the global topology of the

978-1-4673-0775-8/12/$31.00 ©2012 IEEE

network. Besides, this dependency leads to performance quite sensitive to the boundary noise. More severely, since they exploit the network's 2D geometric properties, this hinders their applicability in 3D sensor networks. In 3D space, we are aware of one partial solution proposed in [21] based on bottleneck identification. A parameter named injectivity radius is calculated by each boundary node. The purpose of this parameter is to measure the narrowness of the nearby boundary area, so as to identify the undesired bottlenecks in a 3D sensor network. These bottlenecks are then used to partition the network boundary, and the non-boundary nodes are grouped finally. Due to the hallmark of bottleneck identification, this algorithm is inapplicable to networks without bottlenecks (see the networks in Fig. 7). Finally, it does not work for 2D networks.

2D/3D segmentation algorithms have been studied in the computer vision and graphics area [14], which target at continuous shapes and use centralized solutions. Despite the resemblance to those works, the problem we strive to address is more challenging because of the nature of a distributed sensor network. First, the random deployment of sensor nodes makes the geometric objects (e.g., holes) do not necessarily follow the properties of their counterparts in a continuous space. The nature of the random deployment also makes impractical to manually identify convex/concave regions during deployment or extract a graph of the network. Second, in practice sensor nodes may have no knowledge of location information. As a result, inter-node distance is often estimated by shortest path hop count, whose measurement accuracy is adversely affected by network concavities. Third, since the sensor network is discrete, it is impractical to decompose a sensor network into strictly convex regions due to the boundary irregularity. It is more meaningful to decompose the network into approximately convex regions [19], [22]. However, how to provide a bound of convexity deviation is not straightforward.

B. Our Contributions

This paper brings a new view to segmentation from a Morse function [3] perspective, bridging the convex regions and the Reeb graph of a network. Accordingly, we propose CONSEL for CONnectivity-based SEgmentation in Large-scale 2D/3D sensor networks. In CONSEL, several boundary nodes first perform flooding to construct the *Reeb graph*. The ordinary nodes then compute *mutex pairs* locally, generating a coarse segmentation layout. Next the neighbor regions that are not mutex pair are merged. Finally, by ignoring mutex pairs that lead to small concavities, we provide a configurable bound for the subnetworks' deviation from convexity. CONSEL is desirable compared with previous solutions: (1) it works for both 2D and 3D sensor networks; (2) it only relies on network connectivity information; (3) it provides a bound of convexity deviation. CONSEL is distributed as no centralized operation is required, and is scalable as both its time complexity and its message complexity are linearly proportional to the network size. Extensive simulations show that CONSEL works well in the presence of holes and shape variation, always yielding appropriate segmentation results.

The remainder of the paper proceeds as follows: Section II outlines the background of Morse functions and Reeb graph. Section III is devoted to a description of CONSEL algorithm. We evaluate our CONSEL in Section IV. Finally, Section V concludes the paper.

II. Preliminary

A. Morse Functions and Reeb Graph

1) Morse Functions: For a manifold M, a Morse function [3] is a mapping $f : M \to R$, where R is the set of real numbers, which can be considered to be a projection from higher dimensions to one dimensional manifold. That is, the outputs of a Morse function are real numbers.

In this paper, the Morse function f is constructed as follows (see Fig. 2): given an origin node o, for each sensor node p in the sensor field, $f(p)$ is referred to as the hop distance between the node p and the node o. In Fig. 2, all the nodes on the same arc have an equal value of $f(\cdot)$. We refer to $f^{-1}(r_0)$ as the set of nodes whose Morse function values are r_0.

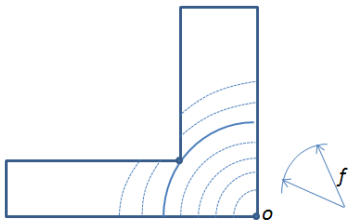
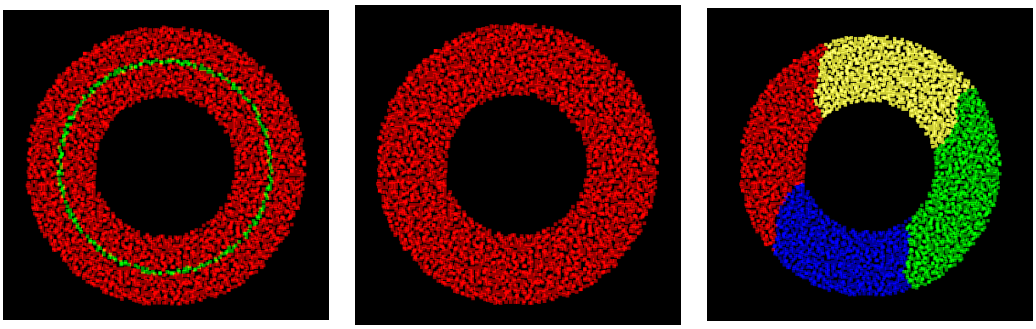


Fig. 2. A simple example of the Morse function.

2) Reeb Graph: Reeb graph has a mathematical foundation in Morse theory. It often has two basic operations: first, we construct a Morse function $f : M \to R$; second, we contract the connected components of the level sets f^{-1} to nodes.

Fig. 3 illustrates the Reeb graph (red nodes and edges) of the network in Fig. 2. The Reeb graph is determined by the changes in the number of connected components of f^{-1}. In Fig 3, the Reeb graph has three vertices, each corresponding to one connected component of the network, which roughly reflects the network's topology. In addition, the lines (defined by a set of connected nodes) of $l(p_1, p_2)$ and $l(p_2, p_3)$ are called *cuts*. The set of a network's cuts corresponding to f is denoted by $L(f)$. For example, the cuts in Fig 3 partition the network into three convex regions.

In general, the number of connected components changes when a concave node exists [3], as shown in Fig. 2. In order to detect the concave parts effectively, we choose to evenly select t origin nodes, $\{o_i | i = 1, \cdots, t\}$ on the outer boundary (see details in Section III-A). We then construct the Morse functions as follows: $f_i(p_j) = \mathbf{dist}(p_j, o_i)$. where $j = 1, \cdots, N$ and $\mathbf{dist}(\cdot, \cdot)$ is the hop distance of two nodes. Clearly, every Morse function is a projection from higher dimension to one dimensional manifold. In essence, some information is inevitably lost. However, as the number of Morse functions increases, the loss of information quickly decreases.



(a) (b) (c)

Fig. 1. A network of 3844 nodes. The average node degree is 16.4. (a) Medial axis result by [22] where nodes in medial axis are shown in dark green; (b) Segmentation result by [22] or [19] where no obvious geometric features on the boundary exist. Existing algorithms generate a single region; (c) Segmentation result by CONSEL where four (approximately) convex regions are generated.

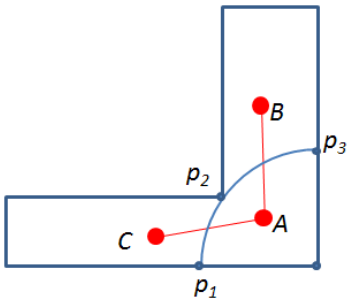


Fig. 3. A simple example of the Reeb graph.

B. Approximate convexity

In theory, a convex region is a part where the line segment between every pair of its inner points lie entirely within the region. However, as shown in Fig. 3, the cuts are arcs instead of straight lines in practice. To address this problem, we give a definition of approximate convexity as this paper aims at partitioning the network into *approximately convex* regions. To that end, we define an ϵ-*straight line*:

Definition 1. *For two nodes p and q, a path from p to q is called an ϵ-straight line if and only if any node in the path is at most ϵ hops away from the straight line between p and q.*

The concept of ϵ-convex regions thus follows.

Definition 2. *For any pair $< p, q >$ in a region, if there exists an ϵ-straight line path from p to q such that all nodes on the path are within the region, this region is called an ϵ-convex region.*

When the network is finally partitioned by a set of cuts into several regions, intuitively each region is approximately convex, or ϵ-convex. As an example, in Fig. 3, let $\mathbf{dist}(p_1, p_3) = d$ and $\mathbf{dist}(o, p_1) = r$. To satisfy the condition in Definition 1, it is obvious that $r - \sqrt{r^2 - d^2/4} < \epsilon$ should hold. That is, $d^2 + 4\epsilon^2 < 8r\epsilon$. One observation is that when $d << r$, this condition holds always.

C. Mutex Pairs

In a continuous space, if an area is convex, then it contains the line segment between every pair of its inner points. This fact motivates us to consider the relation between inner points of a network in the light of Morse functions.

Definition 3. *Given f, for two nodes, p_1 and p_2 and $f(p_1) = f(p_2)$, in the network O, a Morse Path is a path from p_1 to p_2, denoted by $\hat{\texttt{path}}(p_1, p_2)$, such that for any node $q \in \hat{\texttt{path}}(p_1, p_2)$, we have $f(q) = f(p_1)$.*

Definition 4. *Given f, for two nodes, p_1 and p_2 (and $f(p_1) = f(p_2)$), in the network O, if O does not contain a Morse path between p_1 and p_2, we call p_1 and p_2 a mutex pair, denoted by $p_1 \sim p_2$.*

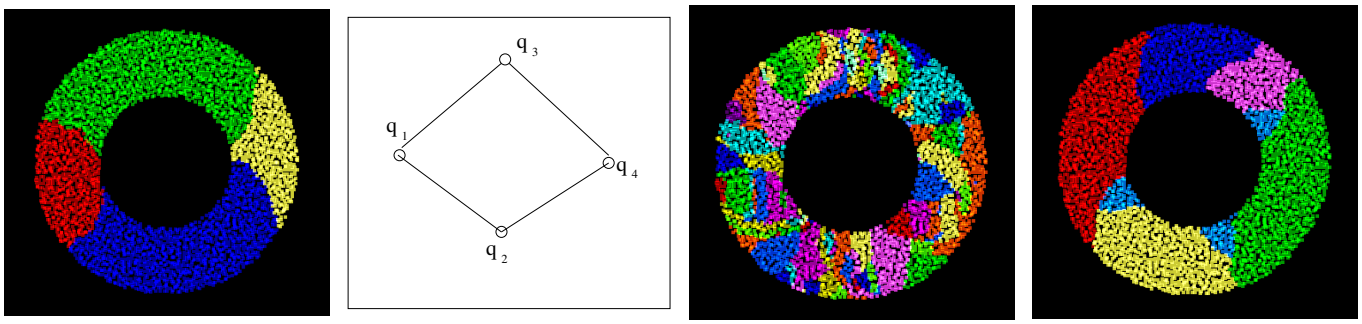
According to Definition 4, in Fig. 3 we can find one node in area B and one node in area C that form a mutex pair. Intuitively, after segmentation, they cannot be in the same region. For two regions A and B without intersection, if there is a mutex pair formed by a node from A and a node from B, regions A and B are called a mutex pair, denoted by $A \sim B$. Note that a node can be regarded as a singleton set, so a mutex pair of regions in fact includes a mutex pair of nodes.

Given the Reeb graph constructed from f, the task of determining a mutex pair is very easy. Since Reeb graph is determined by the changes in the number of connected components of f^{-1}, two connected components (i.e., regions) determined by function value in the range $[r_1, r_2]$ (or two connected components formed by nodes belonging to $f^{-1}([r_1, r_2])$) is a mutex pair. At the same time, the cuts between adjacent nodes of Reeb graph can separate these mutex pairs.

Recall we seek to partition a network into approximately convex parts. We thus have the following definitions.

Definition 5. *Given f, for two nodes, p_1 and p_2 and $f(p_1) = f(p_2)$, in the network O, a δ-Morse Path is a path from p_1 to p_3 to p_4 to p_2, such that: (1) **dist**$(p_1, p_3) < \delta$ and **dist**$(p_4, p_2) < \delta$; (2) $f(p_3) = f(p_4)$; (3) there exists a Morse Path from p_3 to p_4.*

Definition 6. *Given f, for two nodes, p_1 and p_2 (and $f(p_1) = f(p_2)$), in the network O, if O does not contain a δ-Morse path*



(a) (b) (c) (d)

Fig. 4. Step by Step CONSEL algorithm. (a,b) Reeb graph construction; (c) Coarse segmentation layout; (d) Merging result. The final refined result is in Fig. 1(c).

between p_1 and p_2, we call p_1 and p_2 a δ-mutex pair, denoted by $p_1 \sim_\delta p_2$.

Definition 6 gives an approximate mutex pair which allows a small part of non-convex subregions to be merged finally. Overall, ϵ and δ can be considered as two parameters allowing the network operator to control the level of convexity after network segmentation. As such, our algorithm is to generate a set of $(\epsilon + \delta)$-convex regions. That is, CONSEL guarantees a bound for the subnetworks' deviation from convexity.

III. CONSEL Algorithm

In this section we present the implementation details of the segmentation algorithm. It is noted that even in a 2D domain, computing a minimum number of convex components for a polygon with holes is NP-hard [15]. We therefore do not pursue an optimal convex segmentation algorithm. Instead, our goal is to provide a simple and practical algorithm that can be performed in a distributed way.

A. Computing Morse Functions

In the first step, we randomly choose I nodes roughly in the outer boundary of the network. We use a similar technique like the one in [12]: a randomly selected node p performs a flooding to find the farthest node o_1 to p. Thereafter, o_1 initiates a flooding to find the farthest o_2 to itself. Then, o_3 is the node that has the maximum sum of the square roots of the hop counts from the nodes o_1 and o_2. This process continues until I nodes $(o_i, 1 \leq i \leq I)$ are obtained on the outer boundary. Note that this selection phase works well for both 2D/3D networks.

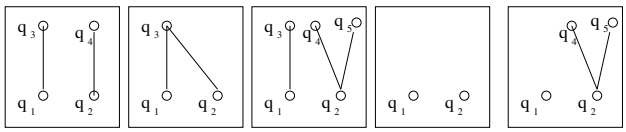
Given I nodes on the boundary, each of them then performs a flooding over the network. The goal of the flooding operations is two-fold. First, after a flooded message from o_i reaches a node p, p records the parent from which it receives the message, as well as the hop count to the node o_i. By doing so, the node p has the knowledge of the Morse function value $f_i(p), 1 \leq i \leq I$ corresponding to $o_i, 1 \leq i \leq I$. Second, we show how to construct the Reeb graph in a distributed way in the next subsection.

B. Constructing Reeb Graph

Recall that Reeb graph is determined by the changes in the number of connected components of f_i^{-1}. The key is to identify the number of connected components at a distance of r hops from o_i, that is $f_i^{-1}(r)$. To do so, a set of randomly selected nodes q_k $(1 \leq k \leq K)$ on $f_i^{-1}(r)$ perform floodings, whose messages contain the information of node ID $ID(q_k)$ and hop count r. The process of selecting these nodes is similar to the landmark selection in [16]: each node on $f_i^{-1}(r)$ will claim it is a landmark with a given probability and broadcast to its neighbors. This kind of flooded messages is only sent to nodes exactly on $f_i^{-1}(r)$. That is, when a node p receives the flooded message, it compares the hop count value r with its own Morse function value $f_i(p)$. If the values are not equal, the message is discarded; otherwise, the node compares $ID(q_k)$ with the landmark ID $LID_i(p)$ (if it has one) it previously received. When $ID(q_k) < LID_i(p)$ the node p will set its landmark ID to be q_k (i.e., $LID_i(p) \leftarrow ID(q_k)$) and then forwards the message to its neighbors except the node from which p receives the message. By doing so, only one landmark with the smallest ID can dominate in each connected component. We call this node a *dominating landmark node*. That is, all nodes in each connected component has the same landmark ID $LID_i(p)$. Without loss of generality, we assume there exist two connected components for $f_i^{-1}(r)$, corresponding to two dominating landmark nodes q_1 and q_2. Under this assumption, for the nodes on $f_i^{-1}(r+1)$, several landmark nodes are found as well but few dominating landmark nodes remain. We consider three cases below.

In the first case, we find two dominating landmark nodes p_3 and p_4 in $f_i^{-1}(r+1)$. Then the edge nodes, that have the different landmark ID with one of its neighbors, will notify the dominating landmark nodes p_1 and p_3 of the simplified local topology graph [1], [9], as shown in Fig. 5(a) (there exists an edge between p_1 and p_3). We consider a simplified local topology like Fig. 5(a), that is, q_1 is with only one edge connecting to dominating nodes in $f_i^{-1}(r+1)$ and q_3 is with only one edge connecting to dominating nodes in $f_i^{-1}(r)$ as well. In this case, a change of the number of connected components does not happen. Therefore, p_3 will broadcast a message to all nodes which have $LID(\cdot) = ID(p_3)$ on

$f_i^{-1}(r+1)$ so that these nodes change their landmark ID, $LID(\cdot) \leftarrow ID(p_1)$. In such a way the simplified topology becomes Fig. 5(d) where p_2 and p_4 become non-dominating landmark nodes.



(a) (b) (c) (d) (e)

Fig. 5. Simplified local topology graph.

In the second case, there are two kinds of nodes on $f_i^{-1}(r+1)$. One kind has landmark ID $ID(p_1)$ and also one of its neighbors has the landmark ID $ID(p_3)$. The other kind of nodes have landmark ID $ID(p_2)$ and also one of its neighbors has the landmark ID $ID(p_3)$. This case is shown in Fig. 5(b). In this case, a change of the number of connected component takes place. Intuitively new vertices (for simplicity, q_k is considered as the vertex) should be added to the Reeb graph like in Fig. 3 where one vertex represents one region. Often newly generated vertices are introduced to the Reeb graph of the network. In addition, two edge are generated on the Reeb graph according to Fig. 5(b).

The last case, shown in Fig. 5(c), is similar to the second one. In this case, there is a change of the number of connected component as well. At least two dominating landmark nodes are connected to q_2. The simplified topology becomes Fig. 5(e) and q_3 becomes a non-dominating landmark nodes. Also, two vertices and two edges are added to the Reeb graph.

Besides the above-mentioned three cases, to guarantee the bound for the final subnetworks' deviation from convexity, the condition of $d^2 + 4\epsilon^2 < 8r\epsilon$ should be satisfied, as we mentioned in Section II-B. To that end, when this condition does not hold for a connected component in $f_i^{-1}(r_0)$ where d is its diameter and here $r = r_0$, this component is subject to be partitioned. Specifically, we find out two nodes, represented by p' and p'', who have the longest path length in $f_i^{-1}(r_0)$ and thus those nodes closer to p' than p'' belong to one region and the rest belong to another region. The condition will be examined again for each region until it is satisfied. This case is then processed similar to Fig. 5(b).

Finally all dominating landmark nodes send their local simplified topology back to the origin o_i. After this phase, the origin o_i has the full picture of Reeb graph based on the Morse function f_i. Fig. 4(a) and Fig. 4(b) show the result of the Reeb graph after the flooding for a Morse function. In Fig. 4(a), the nodes marked with the same color are in the same region, corresponding to one vertex in Reeb graph.

It is noted that we have randomly choose I nodes as origin nodes. Therefore, besides the Reeb graph shown in Fig. 4(b), there are $I-1$ additional Reeb graphs obtained in a similar way since a Reeb graph is generated every time an origin node has performed a flooding operation.

C. Computing Mutex Pairs and Coarse Segmentation

Given the Reeb graph constructed from f_i, the task of computing mutex pairs is very easy. Since the Reeb graph is determined by the changes in the number of connected components of f_i^{-1}, two distinct connected components (regions) determined by function value range $[r_1, r_2]$ (one region composed of nodes in dark green and the other in dark blue in Fig. 4(a), or two distinct connected components represented by q_2 and q_3 in Fig. 4(b)), is a mutex pair. At the same time, the cuts between adjacent vertices of the Reeb graph can separate these mutex pairs.

As mentioned in eariler, all nodes in the same connected component on $f_i^{-1}([r_1, r_2])$ should have been labeled the same landmark ID. For simplicity, these nodes record their landmark ID as $ID(q_k)$ and the dominating landmark q_2 records its mutex pair dominating landmark ID $ID(q_3)$ (here we use Fig. 4(b) as an example). Since I origin nodes are used, each node p records I landmark IDs accordingly.

The subregion recognition/partion works as follows. The basic idea behind coarse segmentation is that all nodes with the same I landmark IDs should be in one subregion. For simplicity, we use the network in Fig. 4(a) where $I = 8$, as an example, in which 32 dominating landmark nodes are obtained. Here each node p maintains 8 landmark IDs $LID_1(p), LID_2(p), \cdots, LID_8(p)$ (corresponding to $o_1, o_2, \cdots, o_8$).

Let these dominating landmark nodes be $q_1, q_2, \cdots q_{32}$. Each of them performs flooding with its ID as subregion ID, denoted by $SID(q_k)$ ($SID(q_k) = ID(q_k)$), as well as its 8 landmark IDs $LID_1(q_k), LID_2(q_k), \cdots, LID_8(q_k)$. Hereinafter q_k is called a *representative node* in a subregion if and only if $ID(q_k) = SID(q_k)$. When no confusion occurs, we use the term *the region* $SID(q_k)$ to refer to the node set of a convex region. The term *the node* $SID(q_k)$ is referred to as the representative node q_k in the region $SID(q_k)$. When a sensor node p receives a flooded message, it compares the landmark IDs with its own $LID_1(p), LID_2(p), \cdots, LID_8(p)$. When they are equal, p simply discards the flooded message. If p has the same landmark IDs with the message, but it has already had one subregion ID ($SID(p)$) and $SID(p) < ID(q_k)$, p still discards the flooded message. Otherwise, it will update its subregion ID ($SID(p) \leftarrow ID(q_k)$) and then forwards the flooded message to all the neighbor nodes except the message's transmitter.

It is noted that not all 32 dominating landmark nodes will become the representatives of subregions because some of them may have the same eight landmark IDs. In addition, some subregions may have no representative nodes since they may not include a landmark node. To address this problem, each of the rest nodes, say node p', that still has not been assigned a subregion ID, will asynchronously claim itself as a representative node in its subregion (and its subregion id is $SID(p') \leftarrow ID(p')$), and then broadcast to all the neighbors that have the same eight landmark IDs. When a node q receives this message, it again compares $ID(p')$ and $SID(q)$. Only

when $ID(p') < SID(q)$, the node p updates its subregion ID with $SID(q) \leftarrow ID(p')$ and forwards this message containing the subregion id $SID(p)$ accordingly.

Finally, every node has one subregion ID ($SID()$) and within each subregion, one node is selected to represent this subregion. The representative node $SID()$ maintains the information of I landmark IDs in this subregion as well as I Reeb graphs to identify the mutex pair later. All subregions in turn are the result of the *coarse segmentation* of the sensor network. Fig. 4(c) shows a result of coarse segmentation of the network where 65 subregions are obtained.

D. Merging Subregions

Often there are many subregions separated by cuts of Morse functions. For instance, 65 subregions are obtained in Fig. 4(c). Since each candidate cut can satisfy some mutex pairs, to achieve segmentation, we only need to select some cuts to satisfy all mutex pairs.

This process of cuts selection is called merging subregions. Each representative node in the subregion maintains a landmark ID list. The representative node, say q_1, sends a message, containing this landmark list $LID_1(q_1), \cdots, LID_I(q_1)$, to another representative node, say q_2, in its neighboring subregion. If q_2 does not find any mutex pair in the two landmark list, it replies a positive message to q_1 to allow the merging of $SID(q_1)$ and $SID(q_2)$. The merging process is quite simple: q_2 will broadcast a message to ask all nodes in its own subregion to update their subregion IDs, that is, $SID() \leftarrow ID(q_1)$. In addition, q_1 will update its landmark list to include q_2's landmark list. This process will continue until no more neighboring subregions can be merged. The result of this step is shown in Fig. 4(d) where 8 regions are finally generated.

E. Refinement

Generally for a network, its mutex set is the set of all mutex pairs that are used to generate strictly convex areas. However, as we mentioned earlier, by ignoring mutex pairs that lead to small concavities, it can generate significantly fewer partitions.

To that end, we exploit the definition of δ-mutex pair, mentioned in Section II-C. Based on the result given in the previous subsection, shown in Fig. 4(d), two regions that are not δ-mutex pair can be merged together. Fig. 1(c) shows the result after the refinement.

This process works as follows. Each representative node q first finds out all mutex pairs with its neighboring subregion. Without loss of generality, we assume $LID_1(q)$ and $LID_1(q')$ are a mutex pair. Motivated by Fig. 3, if there exist two cuts (l_1 and l_2) in $L(f_1)$ such that $l_1 \cap l_2 \neq \emptyset$, $l_1 \cap SID(q) \neq \emptyset$, and $l_2 \cap SID(q') \neq \emptyset$, the subregions $SID(q)$ and $SID(q')$ determine whether a δ-mutex pair exists. To that end, the two cuts l_1 and l_2 perform flooding within the subregions $SID(q)$ and $SID(q')$ so that the remaining nodes have the information of its hop distance to the cuts.

Each node with a landmark ID $LID_1(q)$ ($LID_1(q')$) then checks its hop distance to the cut l_1 (l_2). A negative message is then replied to the representative node $SID(q)$ when the its hop distance is greater than δ. If both q and q' do not receive any negative reply, no δ-mutex pair exists. In this case, the node q notifies the node q' that these two regions can be merged. That is, all nodes in the region of $SID(q')$ will update their region ID to $SID(q)$. This process will continue until no more neighboring regions can be merged. Fig. 1(c) shows the refined result. We emphasize that as mentioned in Section II, each of the obtained subnetworks is an $(\epsilon + \delta)$-convex region.

IV. Simulations

We have implemented a simulator, conducted a series of simulations on various simulated topologies, and compared with the algorithms used in [19], [22]. In the tested topologies, sensors are deployed with a perturbed grid model where the grid has a width around 4.5 and the radio range of all tested topologies is around 6.8. Most of the tested networks have an average node degree of around 10 for 2D, 30 for 3D. By default, we set the parameter δ to be 5 hops and ϵ to be 2 hops unless otherwise stated. Due to space limitation, we only present some representative results.

Evaluation on 2D networks. We first compare CONSEL with two state-of-art algorithms in [19], [22] designed for 2D sensor networks. Fig. 6 shows the results. Since the previous algorithms mostly rely on the knowledge of boundary, it can be seen that they result in representations with an unmanageable number of subnetworks. The results by [19], [22] are quite sensitive to boundary noise (or boundary deformation), as we mentioned in Section I. In addition, they cannot deal with networks like Fig. 1. In contrast, CONSEL provides a bound of convexity deviation of subnetworks, resulting in appropriate segmentation results.

Evaluation on 3D networks. We did not implement the algorithm in [21] since, based on bottleneck identification, it does not work for most networks in Fig. 7 except for Fig. 7(d). Fig. 7 shows the segmentation results of CONSEL for several 3D networks. We can see that CONSEL works well for 3D networks, always yielding appropriate segmentation results. For example, in the 8-shape network, six convex regions are generated, reflecting the impact of the two holes in the network topology.

Sensitivity to node distributions. We next consider another sensor placement scheme: uniform random distribution. This distribution, compared to the perturbed grid model, results in more randomness of node deployment. We show the result in Fig. 8. With the increased randomness, the segmentation results show no significant difference compared with their counterparts in Fig 6 and Fig. 7. Despite the higher variability of the node distribution, CONSEL partitions the network according to the Reeb graph and mutex pair, which makes itself robust to the variation of the node connectivity.

Sensitivity to the parameter δ. Recall that ϵ and δ can be considered as two parameters allowing the algorithm to control the level of convexity of generated subregions after network segmentation. Since they have similar influence on the convexity deviation, we study the algorithm's sensitivity

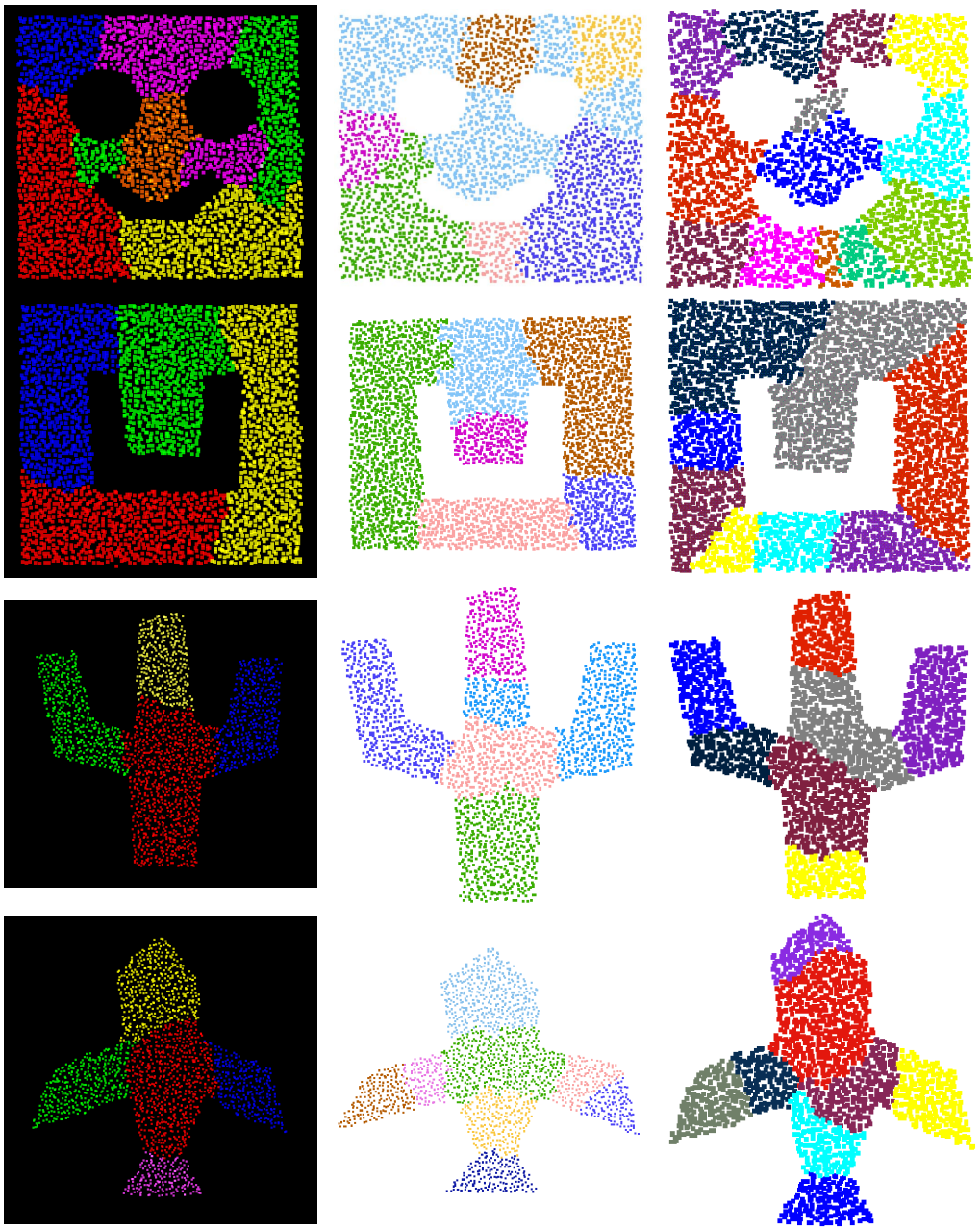


Fig. 6. Comparisons on 2D networks. Columns (from left to right): results using CONSEL, results in [22], results using CONVEX algorithm in [19]. Rows: (1) smile shape, 3045 nodes, avg deg 10.09. (2) single-hole shape, 3700 nodes, avg deg 13.1. (3) Cactus shape, 2,172 nodes, avg deg 8.76. (4) Airplane shape, 1,878, avg deg 8.65.

to the parameter δ. Fig. 9 depicts the segmentation results with varying δ values. It is found that, with increased δ value, more and more approximately convex regions are grouped together. Recall that each of the obtained subnetworks is an $(\epsilon + \delta)$-convex region. This parameter also provides a flexible tool to balance the number of generated subnetworks and their convexity.

V. Conclusion

We have presented CONSEL, a novel distributed and scalable algorithm for segmentation in 2D/3D sensor networks. It is the first solution with both 2D and 3D segmentation capability. This algorithm requires the connectivity information only. In addition, the convexity deviation of subnetworks after segmentation can be bounded. We have demonstrated the effectiveness of CONSEL through extensive simulations.

In the future we will study its uses in applications such as data processing [5]–[7], skeleton extraction [1], [8], [9], routing and localization [20], especially for 3D sensor networks, which have attracted the attention of many researchers.

Acknowledgment

The comments from the anonymous reviewers of INFOCOM helped greatly for this paper. This work was supported

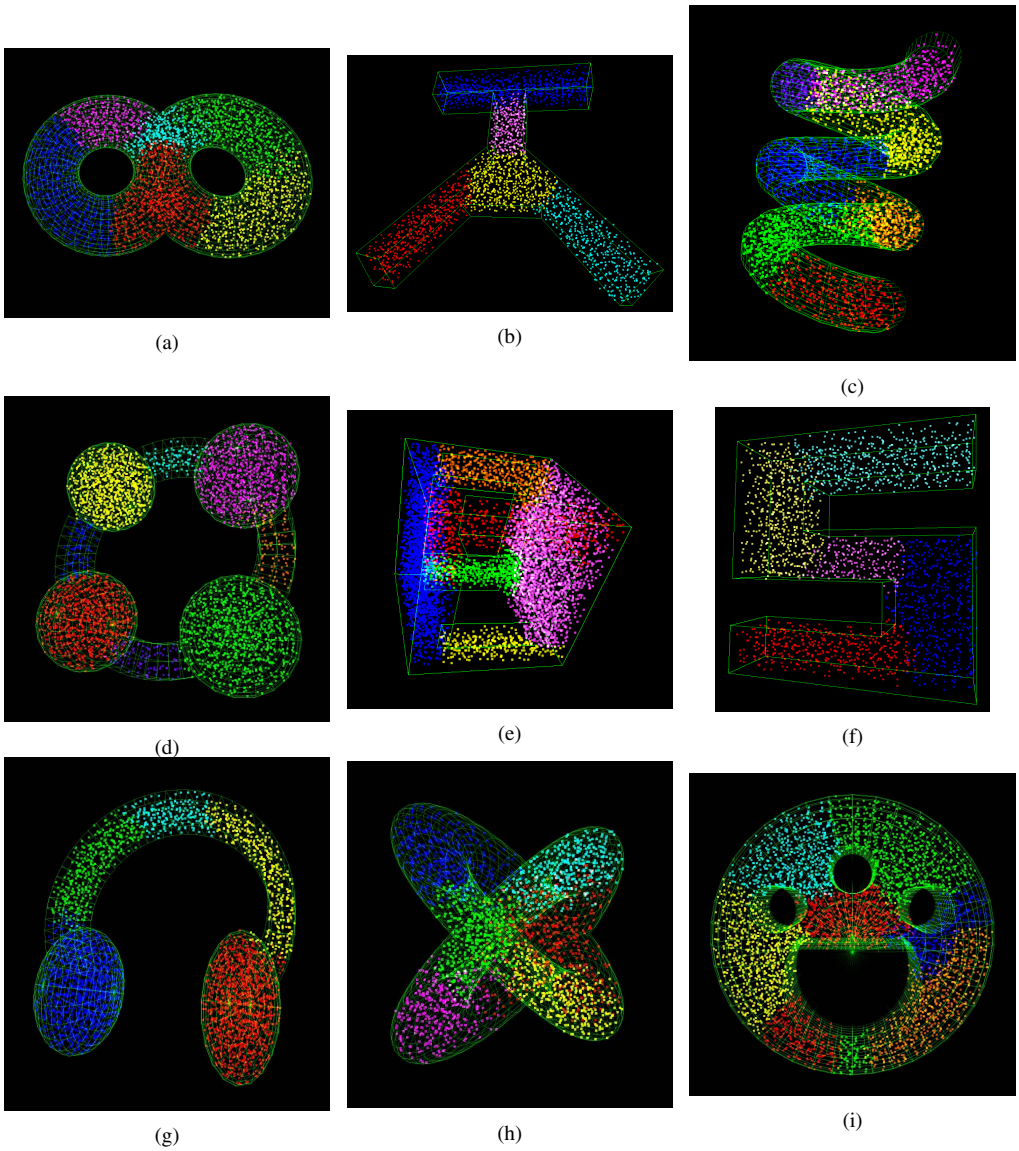


Fig. 7. 3D performance evaluation.(a) 3D-8 shape, 3,486 nodes, avg deg 37.16. (b) Chicago airport terminal 2 shape, 2,502 nodes, avg deg 31.09. (c) 3D-spiral shape, 3,464 nodes, avg deg 31.07. (d) 3D-ring-ball shape, 3,790 nodes, avg deg 33.85. (e) 3D-single-hole shape, 5,681 nodes, avg deg 32.28. (f) 3D-S shape, 2,244 nodes, avg deg 31.38. (g) 3D-headset shape, 3,260, avg deg 31.5. (h) 3D-cross-ring shape, 2,416, avg deg 25.4. (i) 3D-footprint shape, 3,996, avg deg 33.8.

in part by the National Natural Science Foundation of China under Grant 60803115, Grant 60873127, Grant 60903096, Grant 61073147, and Grant 61173120; by the National Natural Science Foundation of China and Microsoft Research Asia under Grant 60933012; by the Fundamental Research Funds for the Central Universities under Grant 2011QN014; by the National Natural Science Foundation of Hubei Province under Grant 2011CDB044; by the Youth Chenguang Project of Wuhan City under Grant 201050231080; by the Scientific Research Foundation for the Returned Overseas Chinese Scholars (State Education Ministry); and by the Program for New Century Excellent Talents in University under Grant NCET-10-408 (State Education Ministry). Dr. Guang Tan's work was supported by the National Natural Science Foundation of China under Grant 61103243.

REFERENCES

[1] J. Bruck, J. Gao, and A. A. Jiang. Map: Medial axis based geometric routing in sensor networks. *Wireless Networks*, 13(6), 2007.

[2] T. K. Dey, J. Giesen, and S. Goswami. Shape segmentation and matching with flow discretization. In *Proceedings of Workshop on Algorithms and Data Structures*, 2003.

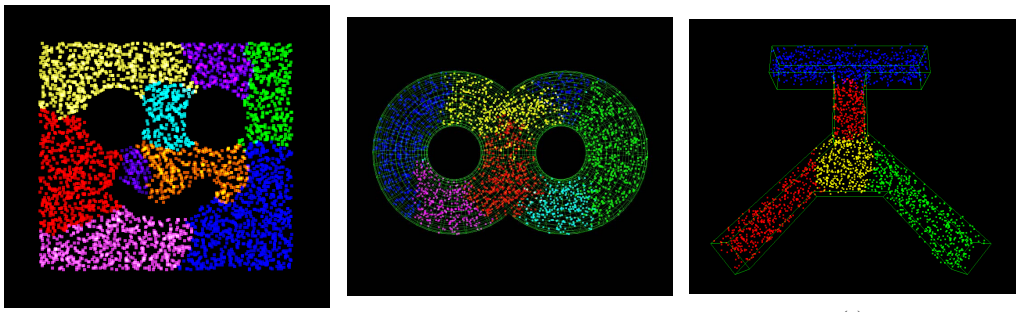
[3] A. Fomenko and T. Kunii. *Topological modeling for visualization*. Springer-Verlag Telos, 1997.

[4] R. Ghrist and A. Muhammad. Coverage and hole-detection in sensor networks via homology. In *Proceedings of IEEE IPSN*, 2004.

[5] H. Jiang, J. Cheng, D. Wang, C. Wang, and G. Tan. Continuous multi-dimensional top-k query processing in sensor networks. In *Proceedings of IEEE INFOCOM*, 2011.

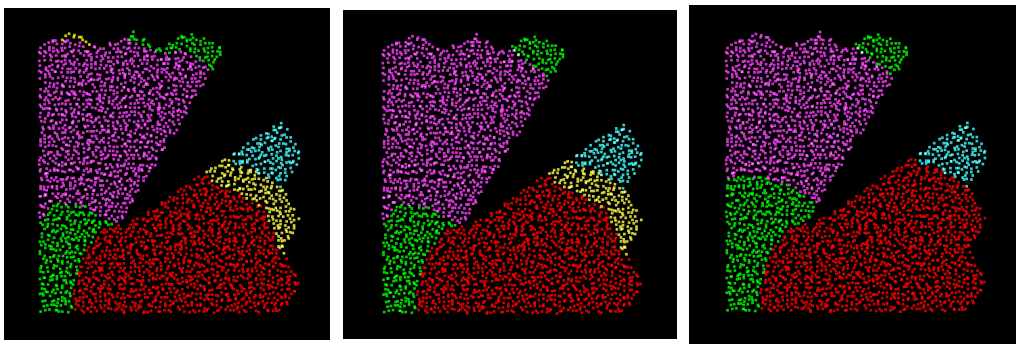
[6] H. Jiang, S. Jin, and C. Wang. Parameter-based data aggregation for statistical information extraction in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 59(8), 2010.

[7] H. Jiang, S. Jin, and C. Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor net-

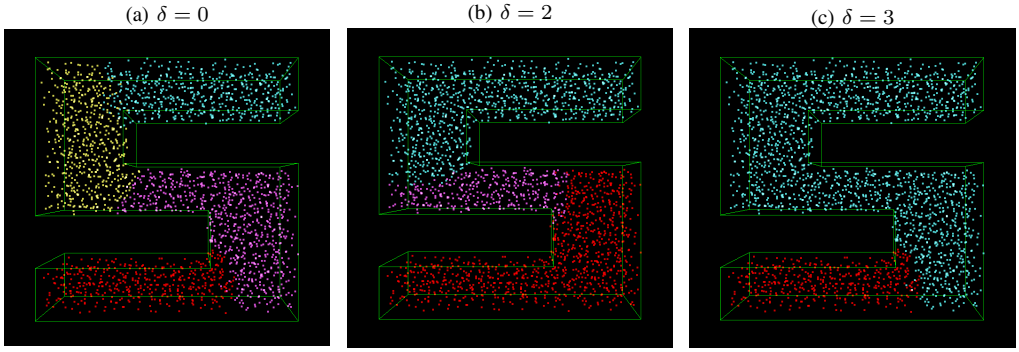


(a) (b) (c)

Fig. 8. (a) 3,388 nodes, avg deg 11.5. (b) 2,859 nodes, avg deg 29.6. (c) 2,585 nodes, avg deg 31.0.



(a) $\delta = 0$ (b) $\delta = 2$ (c) $\delta = 3$



(d) $\delta = 8$ (e) $\delta = 10$ (f) $\delta = 15$

Fig. 9. Segmentation results with varying δ values. The first row shows a network with many small boundary deformations. It contains 2,878 sensor nodes and its average degree is 10.52.

works. *IEEE Transactions on Parallel and Distributed Systems*, 22(6), 2011.

[8] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, Y. Wu, and W. Liu. CASE: Connectivity-based skeleton extraction in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2009.

[9] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, Y. Wu, and W. Liu. Connectivity-based skeleton extraction in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(5), 2010.

[10] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM MOBICOM*, 2000.

[11] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proceedings of ACM PODC*, 2003.

[12] B. Leong, B. Liskov, and R. Morris. Greedy virtual coordinates for geographic routing. In *Proceedings of IEEE ICNP*, 2007.

[13] M. Li and Y. Liu. Rendered path: Range-free localization in anisotropic sensor networks with holes. In *Proceedings of ACM MOBICOM*, 2007.

[14] J.-M. Lien and N. M. Amato. Approximate convex decomposition of polyhedra. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*, 2007.

[15] A. Ling. The power of non-rectilinear holes. In *Proc. 9th Internat. Colloq. Automata Lang. Program*, 1982.

[16] A. Nguyen, N. Milosavljevic, Q. Fang, J. Gao, and L. J. Guibas. Landmark selection and greedy landmark-descent routing for sensor networks. In *Proceedings of IEEE INFOCOM*, 2007.

[17] A. Panangadan and G. S. Sukhatme. Data segmentation for region detection in a sensor network. In *Proceedings of DCOSS*, 2005.

[18] O. Saukh, R. Sauter, M. Gauger, P. J. Marron, and K. Rothernel. On boundary recognition without location information in wireless sensor networks. In *Proceedings of IPSN*, 2008.

[19] G. Tan, M. Bertier, and A.-M. Kermarrec. Convex partition of sensor networks and its use in virtual coordinate geographic routing. In *Proceedings of IEEE INFOCOM*, 2009.

[20] G. Tan, H. Jiang, S. Zhang, and A.-M. Kermarrec. Connectivity-based and anchor-free localization in large-scale 2d/3d sensor networks. In *Proceedings of ACM MOBIHOC*, 2010.

[21] H. Zhou, N. Ding, M. Jin, S. Xia, and H. Wu. Distributed algorithms for bottleneck identification and segmentation in 3d wireless sensor networks. In *Proceedings of IEEE SECON*, 2011.

[22] X. Zhu, R. Sarkar, and J. Gao. Shape segmentation and applications in sensor networks. In *Proceedings of IEEE INFOCOM*, 2007.

[23] X. Zhu, R. Sarkar, and J. Gao. Topological data processing for distributed sensor networks with morse-smale decomposition. In *Proceedings of IEEE INFOCOM*, 2009.