Localised Alpha-shape Computations for Boundary Recognition in Sensor Networks

Marwan Fayed* Hussein T. Mouftah

School of Information Technology and Engineering (SITE), University of Ottawa, 800 King Edward Ave., Ottawa, ON, K1N 6N5 Canada

Abstract

Intuitively, many wireless and sensing applications benefit from knowledge of network boundaries. Many virtual coordinate constructions rely on the furthest set of nodes as beacons. Network edges may also bound routing holes in the network, regions of failure due to environmental effects, or indicate the need for additional deployment. In this paper we solve the edge detection problem locally using a geometric structure called the alpha-shape (α-shape). For a disc of radius $1/\alpha$, the α-shape consists of nodes (and joining edges) that sit on the boundary of the discs that contain no other nodes in the network. In the simplest terms a node decides it is on a network boundary by asking the following question: "Do I sit on the boundary of a disc of radius $1/\alpha$ that contains no other nodes in the network?" We show that using only local communications our algorithm is provably correct. Boundary nodes may further participate to reduce unwanted detail. We show via simulation that our algorithm identifies meaningful boundaries even in networks of low-density and non-uniform distribution.

Key words: wireless, sensor networks, boundary detection, alpha-shapes.

1 Introduction

Context-awareness is increasingly important in wireless and sensor networks. When available, knowledge of position, nearby physical obstacles, or topological features, can be exploited to provide better communication protocols and deployment techniques in resource constrained environments.

* Corresponding author.
Email addresses: `mmf@site.uottawa.ca` (Marwan Fayed*), `mouftah@site.uottawa.ca` (Hussein T. Mouftah).

Intuitively, many pure sensing applications benefit from knowledge of network boundaries ([2,3,8,12,18,21,22,26]). Nodes along the outer edge of the network, for example, are assumed to be the best candidates for beacons in virtual coordinate constructions. Here the assumption is that the finest resolution in coordinates appear using a set of beacons that are furthest apart. Perceived network edges also may bound holes in the network or other regions of interest. Such regions may indicate physical boundaries or node failures due to environmental effects, so that additional nodes may be deployed. In addition, there are applications that benefit directly. KAT [20], whose success relies in part on accurate knowledge of network boundaries, is one such example.

In this paper we solve the edge detection problem locally using a geometric structure called the alpha-shape (α-shape) [5]. The α-shape is used to capture the shape of a set of points in space, and is a generalisation of the convex hull. In addition to geometry-related fields of study such as graphics and computational geometry, α-shapes have been used in molecular biology and particle physics [6]. Alpha-shapes offer real world applicability. They may be weighted to reflect degrees of accuracy relative to network positioning [4], and also extended to three-dimensions [6].

Its use is motivated by the hypothesis that within range of many nodes there exists structural information relevant to the network. For a disc of radius $1/\alpha$, the α-shape consists of nodes (and joining edges) that sit on the boundary of the discs that contain no other nodes in the network. For this initial study we restrict ourselves to network graphs with normalised communication range in two-dimensions.

Our contribution, rather than to suggest a new method, is to identify wireless network boundaries by combining previously unrelated methods. It differs from previous methods in that we investigate what might be achieved if relative information - positions of nodes relative to their neighbours - was known or computable. First, each node constructs a local coordinate system. (Alpha-shape computations are unaffected by translations and rotations in space.) Next, each node computes the Delaunay triangulation of its neighbourhood to find the corresponding α-shape. In the simplest terms a node decides it is on a network boundary by asking the following question: “Do I sit on the boundary of a disc of radius $1/\alpha$ that contains no other nodes in the network?” Finally, any boundary node may request a map of the boundary by transmitting a discovery packet along edges of the α-shape using right-hand rule.

The key to localisation is to select the α-parameter appropriately. In the version of the problem posed in this paper α is selected so that $1/\alpha = 1/2R$, where R is the communication range. Given such an α, the α-shape derived from local computations is provably correct. Even so, the α-shape may expose some unwanted detail that we refine using either of two proposed methods.

We show via simulation that our algorithm identifies meaningful boundaries even in low-density networks. In addition to varying density, we vary topology by generating networks where node locations are selected from uniform, normal, and skewed (Pareto) distributions. We complete our evaluation with a comparison of local α-shaping over dædal topologies presented in [24]. We find that α-shaping produces similarly favourable results with fewer communications and fewer neighbours.

2 Related Work

Our work appears amid a growing body of research on boundary detection. We focus on works that are distributed or localised. Existing work may be classified according to the taxonomy presented in [24] as either geometric, statistical, or topological in nature.

Geometric solutions to the boundary identification problem use the positions associated with each node. Our work falls into this category. To our knowledge the work by Fang *et al.* [7] is the first such work. In it the authors find those nodes abutting any 'hole' in the network as defined by greedy routing techniques. Their methods detect such nodes using local information. A network node is found to abut a local minima if its neighbourhood reflects geometry as defined by a tent-rule. Using the tent-rule, the detection of a boundary node as defined by greedy routing it local; however, a probe is required to identify all remaining nodes along the network edges. In [10] boundaries are identified using a localised convex hull algorithm. Though heuristic in nature, it is found to be resilient to position-estimation error.

Probability distributions underlying network deployments have been used to formulate statistical solutions. One solution, proposed by Fekete *et al.* [11], relies on the idea that nodes close to network boundaries have fewer incident edges in the network graph than internal nodes. The authors use statistical methods to derive suitable thresholds to separate edge nodes from internal nodes using the node degrees. In [23] a similar statistical separation is proposed. Boundary nodes are separated from internal nodes by using a 'centrality' measure which counts the number of shortest paths that pass through a node. A higher centrality value occurs among internal nodes. Statistical solutions generally hinge on uniformly distributed networks and exceedingly high densities. Our method shares in the view that nodes at the boundary exhibit unique characteristics. Unlike statistical methods, our approach is localised, is shown to be resilient to the underlying distribution, and performs well in lower densities environments.

Topological solutions appear in [13,17,21]. Kröller *et al.* propose a combinatorial approach in [17]. It is the only deterministic work of which we are aware to produce correct results without relying on the unit disc graph model (where all communication ranges are normalised). This solution comes at a high cost: It deals with complex combinatorial structures in a distributed manner. Rao *et al.* [21] suggest a single-beacon broadcast solution with a total messaging cost is similar to localised approaches such as ours. Their work begins with a 'hello' message from a single beacon located at least 1-hop from the network edge. A node decides it is on the edge if it finds itself to be furthest from the beacon amongst nodes in its 2-hop neighbourhood. Funke [13] later proposed extensions to a similar idea. The idea relies on the observation that 'rings' of the network, described by hop-distances to a beacon, are broken when encountered by a network boundary. This project was later refined by Funke *et al.* in [14]. Wan *et al.* [24] use topological inference to detect internal boundaries. Their detection method works by identifying the distinct portions of similar paths that span the network.

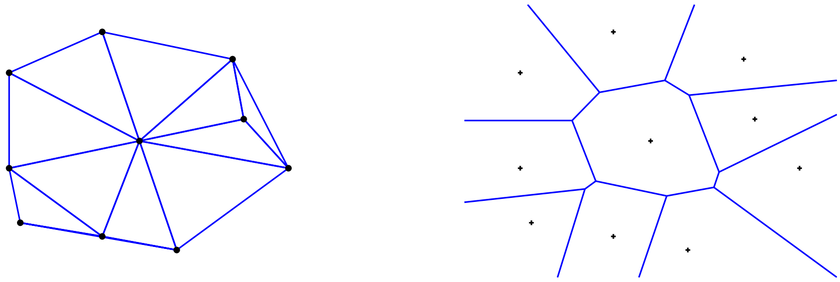
Zhang *et al.* [25] provide a planarization of the network graph that requires neither uniform communication range nor node locations. They propose a progressive construction of trees and their arrangement as bipartite graphs. By planarizing each bipartite graph recursively, the faces of network boundaries emerge as a natural consequence. Finally, somewhat related is the contour tracking project [27]. Here the authors identify the boundaries of a binary event. A binary event may be, for example, a chemical concentration beyond some threshold or a set of grouped targets. When tracking such events there are 'binary' nodes, labelled black and white, with entire neighbourhoods either inside or outside of the events, respectively'. Using limited scope broadcast it is possible to identify the contour consisting of 'grey-area' nodes, nodes with some black neighbours and some white.

3 Preliminaries

Our work exploits the properties of many well known geometric structures. In advance of the presentation of the edge detection algorithm we present some necessary definitions and background for completeness.

3.1 Definitions

Delaunay Triangulation. Given a set of points S the Delaunay triangulation DT_S is one that satisfies the 'empty circle' property, where no point lies inside the circumcircle of any triangle in DT_S. An example of a Delaunay triangulation appears in Figure 1a. It is a super set of the convex hull as well as the minimum spanning tree of S. The number of edges in the Delaunay triangulation is on the order of the number of nodes (ie. $|E| = O(|V|)$ for edge set E and node set V). Of its many properties we are most interested to its relationship to Voronoi Diagrams.



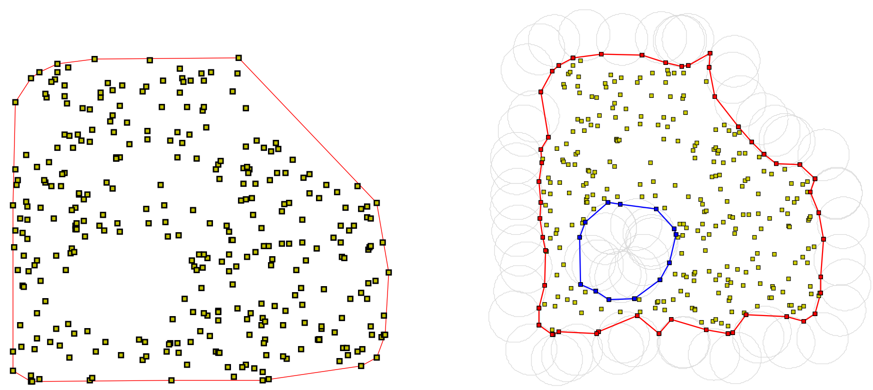
(a) Delaunay Triangulation (b) Voronoi Diagram

Fig. 1. The Delaunay triangulation of a set of points and its corresponding Voronoi diagram.

Voronoi Diagrams. Given a set of points S the Voronoi diagram VD_S partitions the space occupied by S into convex regions $V(p)$ where, for each p in S, any point in $V(p)$ is closer to p than any other point in S. The Voronoi diagram is the dual graph of the Delaunay triangulation. The Delaunay triangulation may be used to compute the corresponding Voronoi diagram in linear time. Referring to Figure 1 we can see both the Delaunay triangulation and the Voronoi diagram for the same set of points.

3.2 α-Shapes

The α-shape provides the foundation of the work in this paper. It is derived from the α-hull, which is a generalisation of the convex hull. To better understand α-shapes we restate the following definitions from the original work in [5] in a manner that better suits our application, and follow with a discussion using examples. Say that we have a set S of points in the plane.



(a) Convex Hull. (b) α-shape.

Fig. 2. The convex and α-shapes of a corresponding set of points.

Definition 1 *The α-hull is defined as the intersection of the complement of all closed discs of radius $1/\alpha$ that contain no points in S.*

We are more interested in a related structure called the α-shape which first requires the following definition.

Definition 2 *A point p in S is said to be α-extreme if p lies on the boundary of a closed disc of radius $1/\alpha$ that contains no other points of S. Two such points p and q that lie on boundary of the same disc are said to be α-neighbours.*

Finally, we may define α-shapes as follows.

Definition 3 *For a set S of points in the plane and $\alpha \geq 0$, the α-shape is the graph whose vertices are α-extreme and whose straight line edges connect α-neighbours.*

Note that as α approaches zero the size of the disc approximates a halfplane. At $\alpha = 0$ the α-hull is identical to the convex hull.

The contrast between structures may be seen in Figure 2. The convex hull of an example set of points appears in Figure 2a. Certainly the convex hull of a wireless network is one way to define its boundary; unfortunately the convex hull fails to reflect numerous features. Observe, for example, that the curve along the upper-rightmost edge is hidden, as is the large circular gap that appears in the lower-left. Furthermore edges of the convex hull, restricted by range constraints inherent among wireless devices, may be impossible to compute.

Using the same set of nodes we compare the α-shape shown in Figure 2b. For demonstration the discs used to produce the α-shape appear using greyed lines. Observe that the revealed shape more accurately reflects the shape of the network (for proper selections of α). The α-shape closely follows variations in the outer-edge. It also reveals the inner gaps. We note that the α-shape can reveals inner as well as outer boundaries, a characteristic missing from the convex hull.

The structures discussed above are well-known, well-studied, and have been extended to 3-dimensions. The algorithm presented in the next section uses these structures to a provide boundary detection localised for wireless and sensor networks.

4 Localised Boundary Detection Algorithm

We consider a deployment of a large sensor network where, initially, nodes may lack any knowledge of their positions. We wish to identify, using only local information, those nodes and links that lie on the network boundaries. We propose a localised algorithm that makes only two assumptions. First, that generated or assigned node IDs are unique within each neighbourhood; second, that distance measurements are available should position information be unknown.

Our method relies on the hypothesis that within the local view of each node there exists some structural information relevant to the network. As discussed in Section 3.2, it is well known that α-shapes, given a proper selection of the α-parameter, reveals a set of nodes and edges that captures the shape of a set of points in a geometric space. We first outline our algorithm and then discuss each step in detail. The correctness of the algorithm is reserved for discussion to Section 5.

(1) (Optional) Each node i constructs a local coordinate system consisting of the nodes $L(i)$ within communication range. Only relative positions are required for the algorithm to succeed. This step is necessary only if position information is unknown.
(2) Each node i for its neighbourhood $L(i)$, constructs the Delaunay triangulation $DT_{L(i)}$.
(3) Given communication range R, select the α-parameter so that the radius r of the disc is $r = \frac{1}{2}R$. Identify 'α-extreme nodes' of $L(i)$ as determined by r. Each node ascertains its boundary status by asking, "Am I α-extreme?"
(4) (Optional) A map of the boundary may be obtained by sending a discovery packets according to right-hand rule along the edges joining α-neighbours.

4.1 Establish a Local Coordinate System

In the first step each node constructs a local coordinate system so that the α_L-shape may be computed in the next steps. Though not the focus of this work, we present this step for completeness. It is described for cases where no a priori position information exist. Should position information be available, this step becomes redundant and may be omitted.

Each node in the network, after announcing its presence, begins by sharing with its immediate neighbours a vector of measured distances to 1-hop neighbours. Once received, each node constructs a local coordinate system by placing itself at the origin and, in a depth-first manner, assigns coordinates to remaining neighbours relative to those whose local coordinates have been

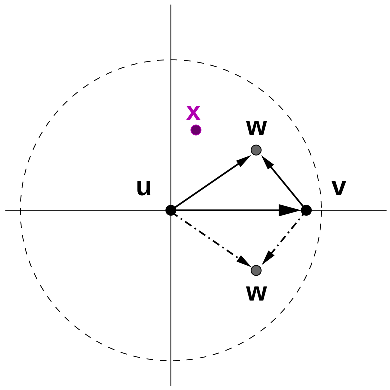


Fig. 3. Node u may generate a local coordinate system, if needed.

established. We demonstrate this idea from the perspective of node u in Figure 3. Node u places itself at the origin of a Cartesian space and sits neighbour v on the horizontal axis. (For our purposes we use the furthest neighbour.) The next node, w, may be placed on either side of the horizontal axis since α-shape computations are resilient to rotations and translations. Remaining nodes are placed similarly and with respect to established coordinates.

Clearly this approach is independent of under- and over-lying protocols. Still, its weakness is its dependence on accurate measurement methods. The viewpoint we take is that location-discovery technologies such as GPS will continue to decrease in size and cost, rendering this argument moot.

Once nodes have discovered their positions relative to their neighbours, each node proceeds to construct the Delaunay triangulation consisting of the nodes in view.

4.2 Local Construction of the Delaunay Triangulation

Construction of the α-shape of a set of points in the plane begins with the construction of their Delaunay triangulation. The communication range restricts us to the unit Delaunay triangulation, where edges longer than the communication range are omitted. Localised constructions of the unit Delaunay triangulation are known not to exist (see [1,15,19]). Despite this fact we show in Section 5.1 that it suffices for our purposes to build only the Delaunay triangulation of the local neighbourhood.

4.3 Boundary Recognition using Local α-shapes

In Step 3 each node i independently determines whether it sits on a network boundary by computing its $\alpha_{L(i)}$-shape, the α-shape of its neighbourhood $L(i)$. Recall from Section 3.2 the definition of α-extreme. From the Delaunay triangulation $DT_{L(i)}$ each node i finds α-extreme points in linear time as follows:

a) Node $p \in L(i)$ lies on the convex hull of $L(i)$; this is the trivial case where node p is α-extreme.
b) Node $p \in L(i)$ is not on the convex hull of $L(i)$; we consider the discs (ie. circumcircles) as defined by the Delaunay triangulation of $L(i)$. Recall from Section 3.1 that the points q which define the convex region V_p enclosing p in the Voronoi diagram, are the centers of the circumcircles touching p in the Delaunay triangulation. Thus any p is α-extreme if $r \leq dist(p, q)$ for any $q \in V_p$. (Edges between α-neighbours are similarly identified.)

Finally, each node asks of itself if it is α-extreme. No node may declare any other node as α-extreme. This restriction avoids many of the pitfalls that plague other localised methods such as [16]. We show in Section 5 that this decision constraint, in addition to our selection of α, is necessary for correctness.

4.4 Mapping the Network Boundaries

We emphasise that, following Step 3, the 'network' has identified all of its boundaries. Collectively, the information stored at *alpha*-extreme nodes constitutes the α-shape of the network for $1/\alpha = r$. Still, it may be advantageous to map the network boundaries. The α-shape, as a subgraph of the Delaunay triangulation, is a planar graph. This fact permits nodes to map network boundaries by routing a discovery packet along edges joining α-neighbours using the right-hand rule : Upon receipt of a discovery packet, an α-extreme node forwards along the next edge in angular order.

The combination of right-hand rule over a planar network graph guarantees the return of a discovery packet to its origin node. In the next section we show that the local α-shape algorithm is correct for certain selections of α.

5 Algorithm Correctness

In this section we demonstrate the correctness of the local *alpha*-shape algorithm. We show that when communication range is normalised, local computation is sufficient and reveals the same nodes and edges as if computation was centralised.

5.1 Local Delaunay Triangulations Suffice

The α-shape of a set of points S may be found using their Delaunay triangulation $DT(S)$ and corresponding Voronoi diagram $VD(S)$. We show by example that, if the communication range is normalised, the events outside of communication range bear no affect on the correctness of local α-shape computations.

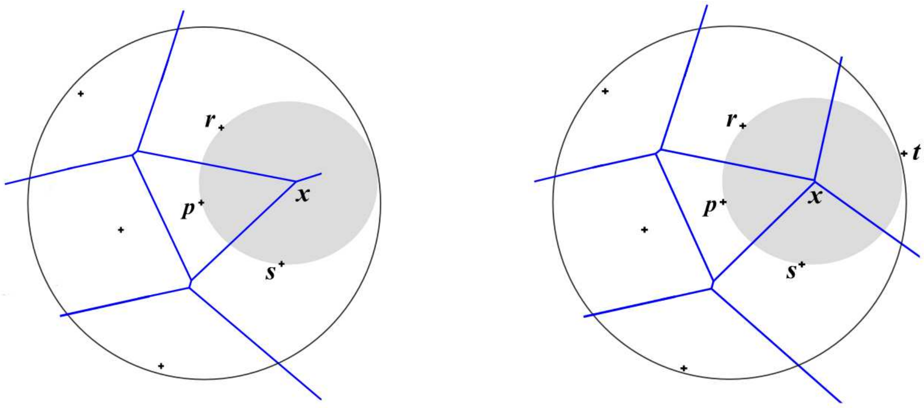


Fig. 4. (a) Local information is sufficient to find α-extreme nodes. (b) Point x in the Voronoi diagram can be no closer to p despite node t outside of range.

We direct our reader to the example in Figure 4, which depicts a 'before and after' scenario of a neighbourhood from the perspective of sensor node p. The large circle represents p's communication range; the greyed disc is the disc used to find α-extreme nodes and edges; the straight lines represent Voronoi diagrams. The Voronoi diagram computed by p is shown in Figure 4a. The intersection x is the center of the circumcircle of prs and reveals that p is α-extreme. We then add a node t just beyond p's view in Figure 4b and show how the Voronoi diagram would change if t was visible to p. Note that x is no closer to p, so p remains $alpha$-extreme. Similar examples may be constructed for any t outside of communication range.

The key to this observation is the bisector property of the Voronoi diagram where any line or point between nodes p and q sits on the line that bisects the space between the points. Next we show that it is possible to ascertain the $alpha$-shape of the network, despite only local information and computation.

5.2 On the Proper Selection of Alpha

Generally speaking, the value of an α-shape rests in the selection of the α-parameter. Edge detection is no different. We ask when, if ever, a locally constructed α-shape might be identical to an α-shape that is constructed in a centralised fashion. For a set of points S in the plane, let $\alpha(S)$ be the set of nodes and edges collected from the centralised α-shape algorithm; similarly we label the set of nodes and edges collected from local operations as described in Section 4 as $\alpha_l(S)$.

Theorem 4 *The sets $\alpha_l(S)$ and $\alpha(S)$ are identical sets.*

Proof. It suffices to consider only the edges in a set since, if any node differs, there must be an edge that differs as well. If we assume that the two sets differ then either or both of the following statements must be true:

a. An edge in $\alpha(S)$ fails to appear in $\alpha_l(S)$.
b. An edge fails to appear in $\alpha(S)$ that appears in $\alpha_l(S)$.

We show both statements to be false. The key is in the selection of α which, in the unit disc graph, is selected so that the radius of the disc $r = 1/2$. When finding the global α-shape and setting α so that $r = 1/2$, only Voronoi neighbours (or their dual, Delaunay edges) whose distance is ≤ 1 may be inserted into $\alpha(S)$. We can satisfy this restriction by searching $UDel(S)$, the Delaunay triangulation with edges greater than 1 unit removed. So, in $\alpha(S)$ we have no edge greater than 1 and only edges whose endpoints sit on an empty circumcircle with $r = 1/2$.

The reverse statement presents a greater challenge since there is no way to construct $UDel(S)$ locally [1,15,19]. Instead we can effectively find all edges that join α-neighbours. Observe in Step 3 of our algorithm that any edge e can only be inserted in $\alpha_l(S)$ by endpoints of e. We label the endpoints u and v. Node u has a complete view of its neighbourhood. This is sufficient information to find incident circumcircles since its radius is set so that its diameter matches the communication range. It is important to note that only incident circumcircles are inserted into $\alpha_l(S)$: Node u is prohibited from making decisions on behalf of v or any other node in its neighbourhood. This idea is demonstrated in Figure 5 in which node u can only insert edges that it knows belong in $\alpha_l(S)$. Finally, the argument is symmetric in that if u inserts uv then v inserts vu. □

Finally, we note that the localised α-hull algorithm is resilient to non-uniform range so long as $r = \frac{1}{2}R_{min}$, where R_{min} is the minimum possible communication range for all nodes. Accurate boundaries emerge so long as R_{min} is sufficiently large to reflect a disc that is able to connect α-neighbours according to Definition 2.

6 Refinement

The selection of disc radius $r = 1/2R$ for communication range R guarantees correctness. Our initial investigation shows that this selection of disc radius exposes some unwanted detail, necessitating further refinement.

We present example misleading geometries in Figure 6. The quadrilaterals' edges represent communicating nodes and boundaries as determined by local α-shaping. The greyed discs used to establish the local α-shape have been

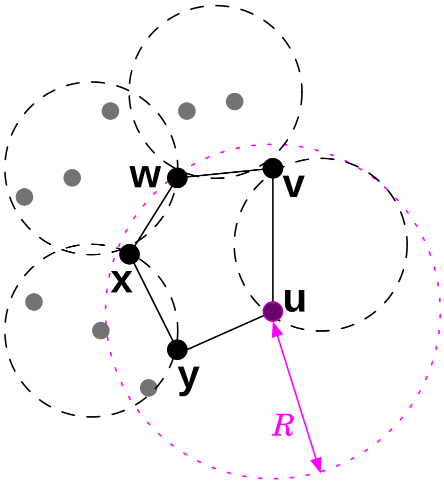


Fig. 5. Node u may insert in $\alpha_l(S)$ only directed incident edges uv and uy.

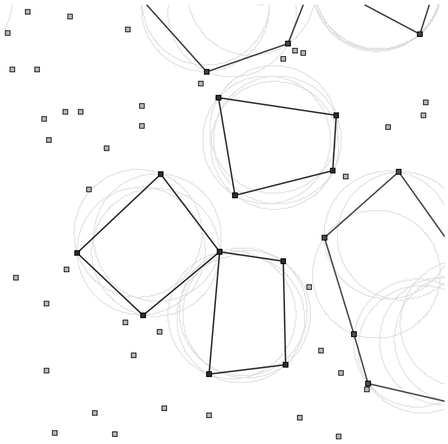
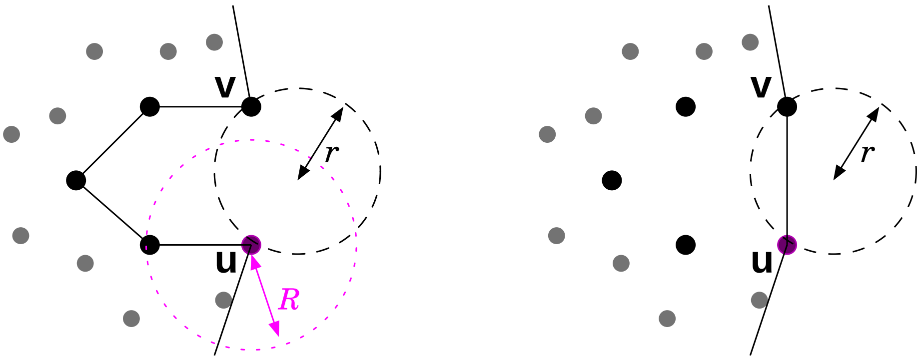


Fig. 6. Nodes in close proximity may expose unwanted detail.



(a) Local α-shape.

(b) Actual α-shape.

Fig. 7. By setting $r > \frac{1}{2}R$, the local α-shape may differ from the actual α-shape of the network.

included for clarity. We can see that sets of nodes located in close proximity may produce boundaries where, effectively, there is none. Unrefined, the local α-shaping process produces boundaries such as those seen in Figure 8. We provide two methods to further refine the α-shape should the default be too fine-grained.

6.1 Increasing the Disc Radius

The first method we investigate is to increase the radius of the disc in Step 3. The outcome is a shape that is more coarse. Under these circumstances the relationship between the α-shapes generated in a local and in a centralised fashion is unclear. The question remains open whether setting $r > 1/2R$, ie. setting the disc diameter strictly greater than communication range, connects local α-edges in a planar configuration amenable to a right-hand-rule mapping.

Increasing the available information to multi-hop neighbourhoods fails to resolve the issue at question. This idea is demonstrated using Figure 7. Referring first to Figure 7a, we present a configuration where the disc diameter is increased beyond the communication range R. Node v is out of the communication range of node u but, unbeknownst to u, inside the disc of radius r. This means that he the segment of the α-shape that would appear if computed centrally, shown in Figure 7b, fails to appear when computed locally.

Hence, by increasing the disc radius an *alpha*-node can successfully determine whether it is located on the boundary; it is unclear if increasing the disc diameter beyond the communication range allows a boundary node to correctly communicate with all other nodes on the same boundary or how the local and global α-shapes might differ. In the following section we present a more effective refinement method.

6.2 Refinement by Omission

Alternatively, network boundaries may be further refined following Step 4 by disregarding those boundaries found to be greater than some pre-determined number of hops in length. Our measurements show that in all but the sparsest networks tested, the length of most boundaries was found to be quite small, measuring 10 or fewer hops. This observation is reinforced by previous study([9]).

Ultimately it is the need to identify boundary nodes versus edges that dictates the refinement method. We present simulations in the next section to compare both methods, and evaluate the local α-shaping process in general.

7 Simulation Results

In the previous section we presented a complete algorithm to identify network boundaries without the need for broadcasting. Here we use a broad set of simulations to evaluate the algorithm performance with respect to density and distribution. We begin with a description of the networks tested.

7.1 Experimental Design

To evaluate the performance of the local α-shaping we simulate networks of varying density, distribution, and topology. Network nodes are distributed in a 200x200 unit space, each node with a fixed range of 8 units. We vary node density by changing the network size. Note that by changing size instead of communication range we can vary the density without affecting the maximum network diameter. Network sizes are 3500, 2500, and 1500 nodes. (In the uniform networks this results in average neighbourhood sizes of $\tilde{1}7$, 12, and 7 nodes.) To obtain results unbiased by isolated nodes we tabulate and experiment over the largest connected component of each network as described by Table 1.

Table 1
Largest connected components in tested networks with 99% confidence intervals.

Initial	Size of lcc		
Network Size	Uniform	Normal	Skewed
3500	3499.9 ± 0.2	3450.8 ± 5.3	3403.8 ± 12.3
2500	2499.8 ± 0.4	2433.8 ± 4.9	2382.2 ± 10.9
1500	1490.0 ± 7.5	1406.7 ± 7.7	1359.0 ± 12.9

Nodes locations are chosen from a normal or skewed (Pareto) distribution in addition to the uniform distribution traditionally used to generate wireless network topologies. Uniformly distributed networks may be sufficient to provide insight yet are poor representations of many real deployments. Normal coordinates are generated with an average of 100 (the center) and a standard deviation of 40. Skewed coordinates are chosen from the Pareto distribution with scale parameter 1.0 and shape parameter 100.5.

7.2 Refinement Phases Compared

We begin our evaluations by comparing local alpha-shaping with and without the refinement phases proposed in Section 6. The outcome of the unaltered localised α-shape algorithm is presented in Figure 8. Small 'empty' pockets appear in the densest networks, growing in number and size in the sparse networks. The less than satisfactory results in Figures 8b and 8c stem from the selected value of the α parameter. Recall that the localised algorithm reveals the same boundaries as the global algorithm so long as the disc diameter is restricted to the communication range. Despite this fact we explore the effect of an increase in α in the first refinement method.

The first attempt at refinement appears in Figure 9 using the same networks as in Figure 8. In these networks the α parameter has been increased by 10% so that $r = 1.10 * \frac{1}{2}r$. Few pockets appear in the denser networks shown in Figures 9a and 9b; unfortunately, the improvement in the sparse network of 1500 nodes shown in Figure 9c, while apparent, is marginal at best. Also, as discussed in Section 6.1, there is no guarantee that planar edges are found when the disc diameter is increased beyond communication range. Routing and mapping, then, becomes a challenge. Still, this approach provides the benefit of remaining entirely local to each node, needing no added communication.

In Figure 10 we show the boundaries revealed by omitting long paths as discussed in Section 6.2. Using this approach, nodes forward a discovery packet to their α-neighbours according to right-hand rule. In our simulations, packets that return to their origin having travelled less than 15 hops are omitted from contention. The improvement is clear. In all but the sparsest network, our algorithm reveals a single network boundary. And in the sparse network shown in Figure 10c only the largest empty regions remain. α-neighbours cooperate to omit unlikely boundaries and produce the most accurate results. The communication required is far less than is required for global broadcasts in comparable methods. We apply this refinement in remaining simulations.

7.3 Distribution and Density of Sensors

In this set of simulations we alter the distribution of nodes in addition to the network density. Distribution parameters are described in Section 7.1. Normally distributed sensor nodes, intended to better approximate aerial deployments, are shown in Figure 11; skewed sensor distributions, intended to better approximate ground projectile deployment, are shown in Figure 12. In all non-uniform simulations presented, α-shapes are shown refined using the mapping method described in Section 6.2. The α-shaping method performs well in all tested networks and, but for increasingly jagged boundaries, seems relatively unaffected by density.

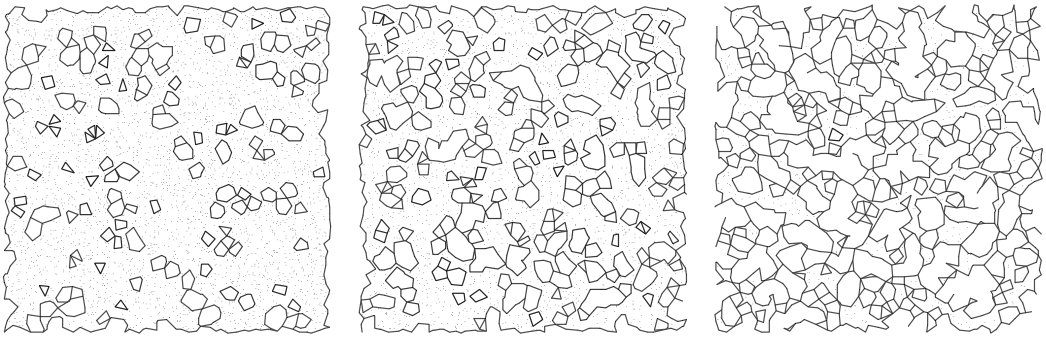
7.4 Additional Examples

Finally, we test local α-shaping using example networks featured in [24]. In that study the authors produced favourable results using topological means needing many network-wide communications. Neighbourhoods in their study ranged from 18 to 22 nodes in size. Our method produced only slightly improved boundaries, and can be seen in Figure 13. However, we emphasise that local α-hull yields slightly more accurate results using far fewer communications and neighbourhoods ranging from 7 to 10 nodes. This is less than half of the size of neighbourhoods in [23].

8 Conclusions and Future Work

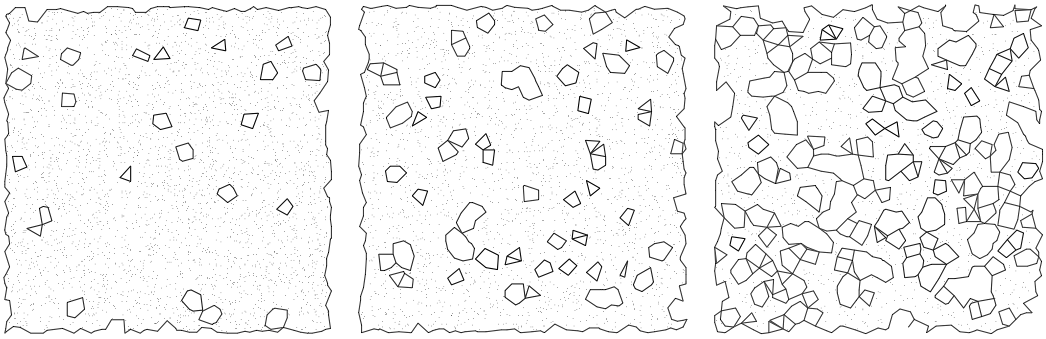
In this paper we have developed an algorithm to identify nodes and links along network boundaries. It is useful to place our work within the context of previous methods.

Edge detection algorithms and protocols have demonstrated considerable potential in the past. Such methods offer the benefit of relaxing the unit disc model, and needing no position information a priori. However, the success of previous methods has relied on the global cooperation of all sensor nodes in the network. The level of cooperation often requires packets to record information revealed over multiple broadcasts. This may be an unacceptable drain



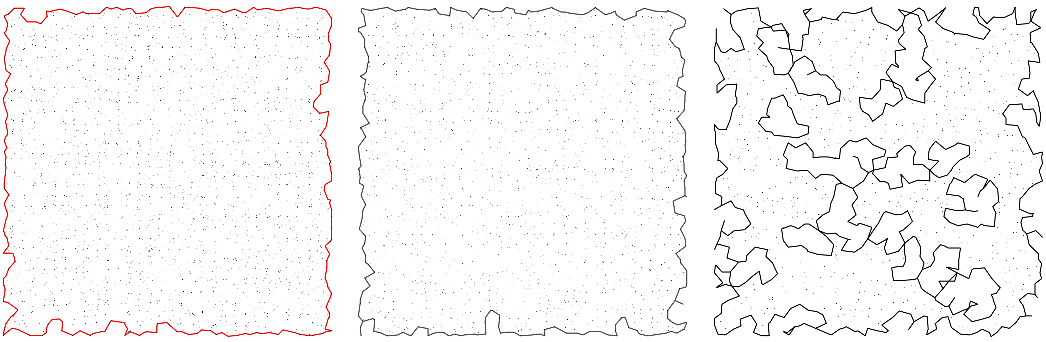
(a) 3500 nodes. (b) 2500 nodes. (c) 1500 nodes.

Fig. 8. Unrefined boundaries determined using local α-shapes; network sizes reflect average neighbourhood sizes of 17, 12, and 7, respectively.



(a) 3500 nodes. (b) 2500 nodes. (c) 1500 nodes.

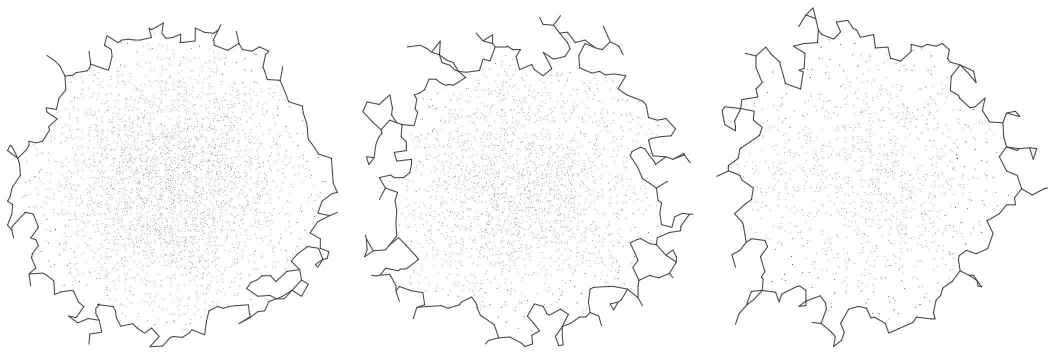
Fig. 9. Disc radius increased by 10%; network sizes reflect average neighbourhood sizes of 17, 12, and 7, respectively.



(a) 3500 nodes. (b) 2500 nodes. (c) 1500 nodes.

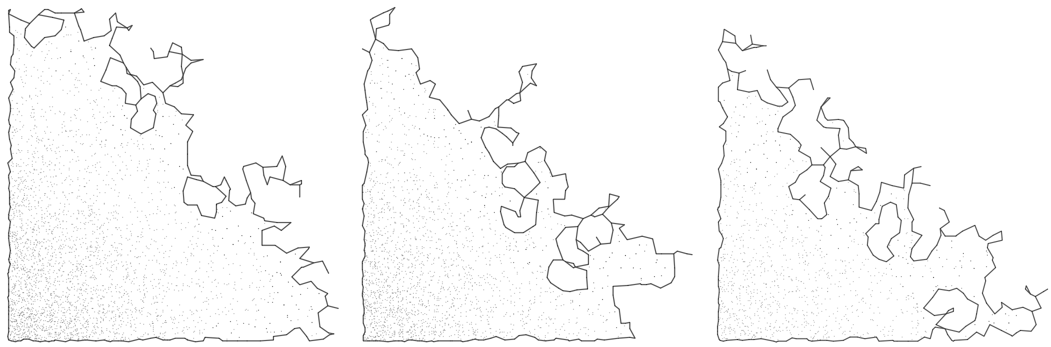
Fig. 10. Boundaries are mapped and omitted if greater than 15 hops; network sizes reflect average neighbourhood sizes of 17, 12, and 7, respectively.

in such an energy constrained environment where channel contention and collision is at issue. Besides this fact a bootstrap node is sometimes assumed to exist, generally at the center or at the edge of the network. The origin of these nodes is unclear. This is restrictive behaviour: many networks, such as those quickly deployed in an emergency, may be unable to tolerate delays in network



(a) 3500 nodes. (b) 2500 nodes. (c) 1500 nodes.

Fig. 11. Results over networks with nodes distributed according to a normal distribution.



(a) 3500 nodes. (b) 2500 nodes. (c) 1500 nodes.

Fig. 12. Results over networks with nodes distributed according to a skewed (Pareto) distribution.

setup or proper placement of bootstrap nodes.

In contrast we assume that partial location information is available or obtainable. Specifically we investigate what might be achieved if relative information - node positions relative to their neighbours - was known or computable. From this standpoint we can investigate geometric options from which we might otherwise be restricted. In this paper we have managed to detect network boundaries by localising the α-shape algorithm; the centralised version of which has been successful in capturing the shape of a set of points in many disciplines. The key is to set α so that the disc in use has a diameter equal to the communication range. Then each node may independently decide if it sits on the α-shape of the network and find the correct incident edges. By using our algorithm and selected α the collection of locally computed nodes and edges provably matches the set of nodes and edges of the α-shape computed centrally.

To validate our approach we tested the local α-shape algorithm in simulated networks of varying density where nodes were distributed according to uni-

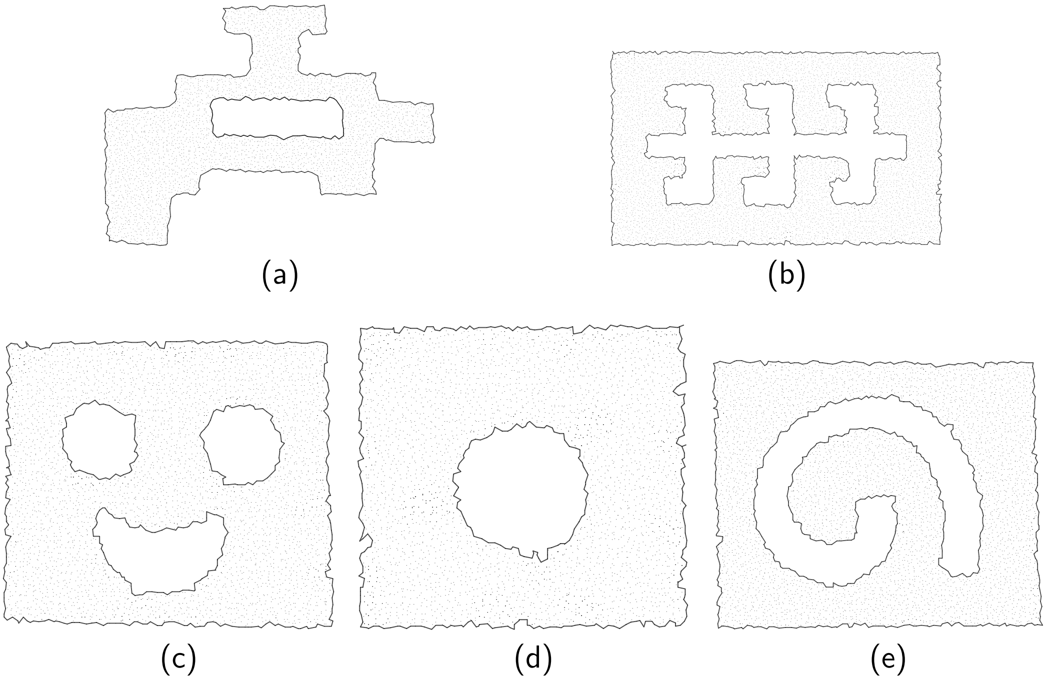


Fig. 13. Dædal examples featured in [24]. They include (a) a building floor plan with 3420 nodes and average degree of 8; (b) a cubicle shaped office space with 6833 nodes and average degree of 7; (c) a happy face with 4050 nodes and average degree 8; (d) a network with 3443 nodes and average degree of 8; (e) a spiral shape with 5040 nodes and average degree of 10.

form, normal, and skewed distributions. We found the algorithm output could be further refined if nodes mapped their boundaries by forwarding a discovery packet according to right-hand-rule; boundaries exceeding some length could be omitted from consideration.

Compared to results from previous studies the local α-shape algorithm performs favourably. Presently, we are looking at statistical methods to resolve inaccuracies associated with position estimation error.

9 Acknowledgements

We thank Yue Wang for providing dædal network topologies. Also thanks to the reviewers whose comments helped to improve clarity. This work supported by the National Sciences and Engineering Research Council of Canada.

References

[1] F. Araújo, L. Rodrigues, Fast localized delaunay triangulation, Springer LCNS Principles of Distributed Systems 3544/2005 (2005) 81–93.

[2] M. Ben-Chen, C. Gotsman, S. Gortler, Routing with guaranteed delivery on virtual coordinates, in: Proceedings of the 18th Canadian Conference on Computational Geometry (CCCG'06), 2006.

[3] Q. Cao, T. Abdelzaher, A scalable logical coordinates framework for routing in wireless sensor networks, Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS) (2004) 349–358.

[4] H. Edelsbrunner, Weighted alpha shapes, Tech. Rep. UIUCDCS-R-92-1760, University of Illinois at Urbana-Champaign, Champaign, IL, USA (1992).

[5] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, Information Theory, IEEE Transactions on 29 (4) (1983) 551–559.

[6] H. Edelsbrunner, E. P. Mücke, Three-dimensional alpha shapes, ACM Transactions on Graphics 13 (1) (1994) 43–72.

[7] Q. Fang, J. Gao, L. Guibas, Locating and Bypassing Routing Holes in Sensor Networks, in: Proceedings of IEEE/ACM Infocom, Hong Kong, China, 2004.

[8] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, L. Zhang, Glider: gradient landmark-based distributed routing for sensor networks., in: Proceedings of the 24th Annual IEEE Conference on Computer Communications (INFOCOM), Miami, FL, USA, 2005.

[9] M. Fayed, H. T. Mouftah, Characterizing the impact of routing holes on geographic routing, in: Proceedings of ICW Systems Communications (SENET), 2005.

[10] M. Fayed, H. T. Mouftah, Localised Convex Hulls to Identify Boundary Nodes in Sensor Networks, International Journal of Sensor Networks (IJSNet) 5 (2) (2009) to appear.

[11] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, C. Buschmann, Neighborhood-based topology recognition in sensor networks, in: Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS), vol. 3121 of Lecture Notes in Computer Science, Springer, 2004.

[12] R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, I. Stoica, Beacon vector routing: Scalable point-to-point routing in wireless sensornets, in: Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05), Boston, MA, USA, 2005.

[13] S. Funke, Topological hole detection in wireless sensor networks and its applications, in: Proceedings of the 2005 joint workshop on Foundations of mobile computing (DIALM-POMC '05), New York, NY, USA, 2005.

[14] S. Funke, C. Klein, Hole detection or: "how much geometry hides in connectivity?", in: Proceedings of the 22^{nd} annual symposium on Computational Geometry (SGC), 2006.

[15] J. Gao, L. Guibas, J. Hershberger, L. Zhang, A. Zhu, Geometric spanner for routing in mobile networks, in: Proceedings of the 2nd ACM international symposium on Mobile ad ho c networking & computing (MobiHoc, 2001.

[16] Y.-J. Kim, R. Govindan, B. Karp, S. Shenker, On the pitfalls of geographic face routing, in: Proceedings of the 2005 joint workshop on Foundations of mobile computing (DIALM-POMC, 2005.

[17] A. Kröller, S. P. Fekete, D. Pfisterer, S. Fischer, Deterministic boundary recognition and topology extraction for large sensor networks, in: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (SODA), New York, NY, USA, 2006.

[18] K. Langendoen, N. Reijers, Distributed localization in wireless sensor networks: a quantitative comparison, Computer Networks 43 (4) (2003) 499–518.

[19] X.-Y. Li, G. Calinescu, P.-J. Wan, Y. Wang, Localized delaunay triangulation with application in ad hoc wireless networks, IEEE Transactions on Parallel and Distributed Systems 14 (10) (2003) 1035–1047.

[20] H. Nakayama, N. Ansari, A. Jamalipour, N. Kato, Fault-resilient sensing in wireless sensor networks, Elsevier Computer Communications 30 (11-12) (2007) 2375–2384.

[21] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, Geographic routing without location information, in: Proceedings of ACM MobiCom, San Diego, CA, 2003.

[22] Y. Shang, W. Ruml, Y. Zhang, M. P. J. Fromherz, Localization from mere connectivity, in: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc), New York, NY, USA, 2003.

[23] S.P. Fekete and M. Kaufmann and A. Krller and N. Lehmann, A new approach for boundary recognition in geometric sensor networks, in: Proceedings 17th Canadian Conference on Computational Geometry (CCCG), 2005.

[24] Y. Wang, J. Gao, J. S. Mitchell, Boundary recognition in sensor networks by topological methods, in: MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking, ACM, 2006.

[25] F. Zhang, A. Jiang, J. Chen, Robust planarization of unlocalized wireless sensor networks, in: IEEE 27th Conference on Computer Communications (INFOCOM)., Pheonix, AZ, USA, 2008.

[26] Y. Zhao, B. Li, Q. Zhang, Y. Chen, W. Zhu, Efficient hop id based routing for sparse ad hoc networks, in: Proceedings of the 13TH IEEE International Conference on Network Protocols (ICNP), 2005.

[27] X. Zhu, R. Sarkar, J. Gao, J. Mitchell, Light-weight contour tracking in wireless sensor networks, in: IEEE 27th Conference on Computer Communications (INFOCOM)., Pheonix, AZ, USA, 2008.