Robust Edge Detection in Wireless Sensor Networks

Christopher J. Mallery  
School of EECS  
Washington State University, WA  
cmallery@eecs.wsu.edu

Muralidhar Medidi  
Dept. of Computer Science  
Boise State University, ID  
MMedidi@boisestate.edu

Abstract—The ability to geometrically represent sensed phenomena within a wireless sensor network can provide a more concise view than enumeration of all nodes identifying a phenomenon. A more concise view of sensed data can reduce the communication and energy costs of data analysis and extend the lifetime of the network. We propose a distributed edge detection technique that identifies connected perimeters for sensed phenomena within wireless sensor networks. The technique operates in arbitrarily deployed wireless sensor networks, such as those containing connectivity holes, and is capable of correctly identifying the perimeters of irregularly shaped phenomena. We implemented our technique and conducted extensive experiments; results show that our technique provides accurate perimeters for all sensed phenomena within a wireless sensor network even in the presence of irregularly shaped phenomena.

I. INTRODUCTION

Wireless sensor networks (WSN) are application-specific wireless ad-hoc networks populated with small, low-cost, resource-constrained immobile nodes equipped with one, or more, external sensors. Regardless of specific application, it is envisioned that WSN be densely deployed over large monitoring areas where the need for efficient and effective collection and analysis of sensor data becomes critical. Traditionally, each individual sensor node forwards its data to a single less resource-constrained location for centralized analysis. However, this approach can cause high network overhead and reduce the lifetime of the network. The potential drawbacks of the centralized collect and analyze paradigm for sensor data analysis makes the development of more advanced data analysis techniques for sensor networks important, which has led to several distinct approaches to solve the problem.

The simplest approach for decreasing the overhead of data analysis in WSN are data aggregation techniques, such as SPIN [?] and LEACH [?]. In general, data aggregation techniques reduce the overhead of data analysis through basic packet operations, such as duplicate packet elimination. A more advanced approach for decreasing data analysis overhead is data fusion. The goal of data fusion is to reduce overhead by having the sensors of the network make intelligent choices about the data itself in order to reduce the obtained dataset quickly, thus minimizing the consumption of valuable resources [?]. Last, and more recent, is the idea of reducing data analysis overhead through the geometric identification of sensed phenomena within a WSN, an approach broadly referenced in the literature as edge detection. A key advantage of identifying geometric representations of a sensed phenomenon is it provides a more concise view than enumeration of all nodes identifying a phenomenon, especially if the phenomenon is large [?]. A more concise view of sensed data can reduce the communication and energy costs of data analysis and extend the lifetime of the network. Geometrically identifying sensed phenomena also provides additional benefits such as the ability to map sensed phenomena onto known geographical features in the deployed region based on geometric shape and to coordinate sleep/wakeup cycles of nodes contained within a sensed phenomena’s identified perimeter.

In this paper, we propose a distributed edge detection technique that identifies connected perimeters for sensed phenomena within WSN. Current edge detection techniques do not attempt to connect identified edge nodes into perimeters, which is the more difficult problem [?]. Our technique is robust in that it operates in the general case under no assumptions about network deployment or shape of phenomena within a network. The basis for our technique is for cooperating groups of location-aware nodes to identify their own outer perimeters using only connectivity information. Groups are comprised of connected nodes with similar sensed data values. Initially each group that has not identified its own perimeter (i.e. a group constructed from nodes not yet bound by any identified perimeter, or captured) cooperatively identifies the group’s convex hull in order to determine the nodes that must exist on the perimeter of the group. The well-known properties of convex hulls provide an intuitive correctness to our technique. The convex hull nodes then connect themselves together, using the paths within the original group of nodes. Since the process of connecting the identified convex hull nodes can leave some nodes uncaptured, the process is repeated on subgroups of uncaptured nodes until all nodes are captured by a perimeter. Once all nodes are captured, the perimeters are merged together to form a single connected perimeter for each group of nodes, one per sensed phenomena. Obviously, if sensed values are ignored our technique is capable of identifying the outer boundary of an entire network.

The rest of this paper’s organization follows. Section II briefly presents related work on edge detection. Section III introduces our approach. Section IV contains performance evaluation and Section V conclusions.

II. RELATED WORK

Chintalapudi and Govindan proposed three different approaches, all loosely based on known image processing techniques, to identify the nodes within the edge of a sensed phenomenon inside a WSN [?]. The authors define an edge
as a region that intersects both the interior and exterior areas of an observed phenomenon. The authors demonstrate the techniques on networks with a single large-scale continuous phenomenon. Nowak and Mitra discuss a technique for detecting an estimated boundary between two regions of relatively homogeneous sensed data [?]. The technique takes advantage of hierarchical quad-trees in order to identify small clusters that estimate the regions in which the boundary between two sets of sensors with differing sensor readings passes. This technique builds upon Chintalapudi and Govindan’s results by considering the existence of multiple large-scale phenomena. Liao et al. have proposed a technique, similar in nature to one of Chintalapudi and Govindan approaches, that not only identifies the nodes on the edge of phenomena, but also the nodes outside the phenomena that border the phenomena’s edge. [?].

Related to the field of edge detection is that of boundary recognition, in which the boundaries of connectivity holes are identified within wireless sensor networks. Boundary recognition techniques can be classified into one of three categories: geometric, statistical and topological [?]. Geometric-based techniques use location information to detect holes in connectivity. Martincic and Schwiebert have proposed a geometric-based technique for identifying nodes on the outer perimeter of a WSN using valid enclosing cycles to differentiate internal nodes from nodes on the perimeter [?]. Fang et. al propose two face routing strategies, one that pre-builds routes around holes in a WSN using Delaunay triangulation and a second simple greedy method for determining boundary cycles in the event that a transmission gets stuck at a node [?], [?].

Statistical methods for boundary detection make assumptions about the statistical properties of a WSN deployment in order to detect holes, in addition to other common assumptions about WSN. Fekete et al present a boundary recognition technique that makes a statistical assumption regarding the degree of nodes on boundaries versus those in the interior of the network [?]. In order to handle different possible boundary shapes, the technique dynamically selects an appropriate threshold for the degree of a node in order to differentiate between a boundary node and an interior node. Another statistical approach towards boundary recognition, also proposed by Fekete et al, computes the “restricted stress centrality” of a vertex, which is the measure of the number of shortest paths of bounded length that go through a vertex [?]. Similar to the first technique, nodes in the interior tend to have a greater centrality than nodes on the boundary. In order to ensure accuracy, both of these techniques require a uniform deployment and an average node degree greater than 100. Bi et al. [?] present another statistical method that identifies boundary nodes because boundary nodes typically have smaller degrees than their 2-hop neighbors do.

Topological methods for boundary detection use only network connectivity information to determine boundaries. Ghrist et al. propose a centralized algorithm that uses homology to detect holes in a WSN for the purpose of determining insufficient coverage area [?]. An algorithm for boundary detection that searches for combinatorial structures called flowers and augmented cycles has been proposed by Kröller et al. [?]. This algorithm does not require the assumption that transmission ranges are perfect disks, instead requiring that the communication graph be a quasi-unit disk graph. Funke and Klein have developed a heuristic for identifying the nodes located on the edge of the network using only connectivity information [?]. The approach constructs hop-based isocontours from a single root node and identifies where the contours break. Wang et al. have developed another approach for boundary recognition using only connectivity information [?]. It builds a shortest path tree from a root node and then identify adjacent nodes whose least common ancestor in the shortest path tree is farther away than it should be, thus identifying a connectivity hole. More recently Fayed and Mouftah developed lev, a convex hull-based technique to accurately identify the nodes on the outer perimeter of a WSN [?].

Our technique, unlike other edge detection techniques in the literature, connects identified edge nodes together into meaningful boundaries. It also operates effectively in networks containing multiple phenomena of arbitrary size and shape. Lastly, by taking advantage of location information to ensure the construction of accurate perimeters.

III. Approach

One necessary assumption for the correct operation of our technique, which is also the case for Chintalapudi and Govindan’s approach [?], is the assumption that every node knows its own position. On the other hand, we do not assume that the communication graph of the WSN follows the unit-disk graph assumption, instead relying only on actual node connectivity to determine which nodes can communicate directly. While the need for nodes to know their own positions within the network could increase the cost of deploying the network if every node was required to be GPS-equipped, our technique only requires a relative coordinate system. A relative coordinate system for a WSN can be determined dynamically after deployment, with no need for any GPS equipment, by taking advantage of one of many relative localization techniques, such as SPA [?], MDS-MAP [?] or CBL [?].

We now provide an outline of our edge detection technique and then elaborate on these steps.

1) Collect all nodes with the same sensed value together into connected groups.
2) Identify the convex hull of the each identified group.
3) Connect each group’s convex hull nodes together using routes consisting only of nodes and edges within the group.
4) While there are uncaptured nodes remaining, repeat steps 1-3 on increasingly smaller groups of nodes.
5) While there is more than one perimeter per initial group, merge the perimeters together in the reverse order they were created.
A. Building the initial groups

The first step of our technique is to collect sensors with the same sensed value into groups, one group per phenomena within the network. All constructed groups maintain the following property: two arbitrary nodes with the same sensed value $\alpha$, are placed into the same group if communication is possible between the nodes by only traversing edges (i.e. communication links) between nodes with sensed values of $\alpha$. Obviously, the discretization of real-valued sensor data is required in order to have a meaningful definition of “the same.” Fig. 1 shows the initial groups, surrounded by the true boundaries of the sensed phenomena, for both a single phenomenon and a multi-phenomena topology.

B. Identifying the convex hulls

Once grouping is complete, the groups identify the nodes that compose their convex hulls. The identification of the convex hull is easily done by the group leader, since it is aware of the position of every node in the group, using the Graham scan algorithm \( ? \). The group leader than notifies the convex hull nodes of their status as well as the identity of the convex hull node immediately following them in an angular ordering of all the convex hull nodes. Fig. 2 shows the identified convex hull nodes for a single phenomenon and multi-phenomena topology. At first, it seems that identifying a convex hull provides less information than simply identifying an angular ordering of all the external perimeter nodes, with something like $lcv$ \( ? \). However, identifying the convex hull of each group provides several advantages. First, identifying the convex hull of each group ensures robustness in the presence of one or more connectivity holes within a phenomenon. In the presence of one or more connectivity holes, an external perimeter detection technique can identify one or more nodes on the edge of a connectivity hole as an external edge node and with no straightforward way to identify false positives can lead to anomalies in the connected perimeter. The second, and most important, ability gained through the identification of each group’s convex hull is the ability to operate on arbitrarily shaped phenomena. In the case of a convex phenomenon, and even some concave phenomena, an angular ordering of all external nodes will provide the same, if not more, information as the convex hull. However, in the case of an arbitrarily shaped phenomenon it is possible that a traversal of the external nodes actually changes directions. In this case, connecting the identified external nodes in angular order will connect the perimeter together incorrectly, whereas connecting the convex hull nodes will correctly navigate the irregularity in the perimeter.

C. Constructing the initial perimeters

The convex hull nodes, identified in the previous step, are likely not within communication range, therefore the convex hull nodes are connected into perimeters using a traditional ad hoc routing algorithm. Once identified, the group leader notifies the entire group of the perimeter and all nodes in the group use this information locally to determine if they remain uncaptured and in turn notify the group leader only if captured. The group leader knows it remains incomplete until it has received captured notices from all nodes in the group, in the case of the newly uncaptured nodes this notice will not come until after merging in Step 5. The intuitive notion is that the connection of the convex hull nodes will result in final perimeters for all sensed phenomena, and in some cases, it may. However, due to the possibility of multiple paths and/or irregularities existing between the convex hull nodes, it is likely that one or more nodes remain uncaptured by the perimeter(s). Fig. 3 shows the constructed initial perimeters for a single phenomenon and multi-phenomena topology, in addition to the groupings of nodes that remain uncaptured. For clarity, enlarged black squares, in the figure, identify groups with only a single uncaptured node. Now any of the remaining uncaptured nodes need assimilated into the initial perimeters, which is the sole purpose of the next two steps.
not assuming that communication follows a unit-disk graph, arise. Making the assumption that a WSN is connected, but makes merging trivial; however, a specific non-ideal case can two newly created edges crossing each other. This ideal case with at least two nodes on the inner perimeter, without the case is that the outer perimeter can communicate directly to merge into a polygon inner perimeter. The ideal merging always polygons. Therefore, outer perimeters only ever need polygon perimeter (i.e. at least three other nodes had to form the actual merging task.

D. Capturing the remaining uncaptured nodes

If no nodes remain uncaptured after Step 3, this and the following step never execute. However, if uncaptured nodes remain, Steps 1, 2 and 3 execute on the remaining uncaptured nodes, ignoring any communication path across the previously identified perimeter(s). Steps 1, 2 and 3 repeat, on an increasingly smaller number of nodes, until there are no nodes that remain uncaptured (i.e. all nodes are contained within some identified perimeter). In general, connections between convex hull points are extremely close to the true perimeter and very few nodes are left uncaptured. Even when nodes actually remain uncaptured they tend to be single isolated nodes or, less often, very small groups of nodes, neither of which require much work to capture into a new perimeter. We have found that only two additional repetitions are required in order to capture all nodes in most topologies. Fig. 4 shows the single phenomenon and multi-phenomena topologies after the capturing of all nodes. For clarity, black squares identify perimeters composed of only a single node.

E. Merging constructed perimeters

Once no uncaptured nodes exist, we are left with the task of merging a set of perimeters, obtained in the previous step, such that one perimeter exists per sensed phenomena within the network. The perimeters are merged together starting with the farthest outside perimeters moving inward towards the initial perimeter identified in Step 3 (i.e. a perimeter is merged into the last perimeter from which its nodes were left uncaptured). The two group leaders of the perimeters that are merging coordinate the actual merging task.

In general, there are three types of perimeters, which we call outer perimeters, that need merging: single nodes, lines and polygons. Outer perimeters merge into what we call inner perimeters. Since nodes can only be left uncaptured by a polygon perimeter (i.e. at least three other nodes had to form the perimeter that left them uncaptured), inner perimeters are always polygons. Therefore, outer perimeters only ever need to merge into a polygon inner perimeter. The ideal merging case is that the outer perimeter can communicate directly with at least two nodes on the inner perimeter, without the two newly created edges crossing each other. This ideal case makes merging trivial; however, a specific non-ideal case can arise. Making the assumption that a WSN is connected, but not assuming that communication follows a unit-disk graph, leads to the possibility that an outer perimeter can only communicate with nodes inside the inner perimeter and cannot communicate with nodes on the inner perimeter itself. In this case, one or more nodes inside the inner perimeter need placed onto the inner perimeter before merging the outer perimeter. Fortunately, it is typical that these nodes are very close to the inner perimeter itself and added easily through short route requests. Once added to the inner perimeter, merging proceeds in the same fashion as the trivial case. Figure 5 shows the final perimeters in both the single phenomenon and multi-phenomena topologies.

While merging uncaptured nodes into the boundaries produced by Step 3 allows us to ensure the construction of accurate final perimeters, they may not be the most compact possible representation of the identified sensed phenomena. Due to the merging process, constructed perimeters may contain unnecessary edge nodes. However, this is easily remedied by identifying and directly connecting non-adjacent nodes on a perimeter that are within one hop, without uncapturing any bypassed perimeter nodes.

IV. PERFORMANCE EVALUATION

In order to evaluate the effect of node density on our edge detection technique, we performed simulations on the same randomly deployed network topology with different average node degrees by varying the communication range of the nodes within the network. The topology consisted of a square region with a single circular sensed phenomenon, similar to the experiment produced by Wang et al. [7]. Fig. 6 shows the results of our approach on randomly distributed nodes with varying average degrees. As expected, our technique provides smoother boundaries, more closely emulating the true boundary of the sensed phenomenon, as the degree increases, since fewer internal nodes are required to connect identified convex hull nodes of the phenomenon. Additionally, our technique is also able to construct accurate perimeters in the event of irregular node placement along the edge of the sensed phenomena, as is shown when the degree decreases. Most importantly, we show that our approach is robust in WSN with low average degree networks. In fact, while the identified boundary may look nothing like the true boundary of the phenomenon, our approach will continue to identify a single connected perimeter for each sensed phenomena within a WSN, even in the presence of average degrees of less than seven, as long as all the nodes identifying a single phenomena.
remain connected amongst themselves. Even in the event that the nodes within a phenomenon are disconnected, our technique will still function; it will simply identify a single connected perimeter for each disconnected group of nodes within the phenomenon.

To further evaluate the effect of node density on our edge detection technique, we also performed experiments involving varying average degree through communication range in a more controlled perturbed grid topology and varying the node density directly in a perturbed grid topology. The goal of the perturbed grid topology, obtained by placing nodes on a grid and then perturbing each point by a small random amount, is to approximate the manual deployment of sensors [?]. All other experiments, as expected, produced similar results as the experiments shown in Fig. 6. Specifically, that our technique provides smoother boundaries, more closely emulating the true boundary of the sensed phenomenon, as the degree increases. However, as expected, our edge detection approach more closely captures the true boundary of the phenomenon in a perturbed grid topology than in a random topology, due to the regularity of node placement provided by the underlying grid.

Also, in order to evaluate the robustness of our technique, in the presence of irregularly shaped phenomena, we adapted two of the more challenging simulation experiments from Wang et al. [?]. Fig. 7 shows the result of our technique on a spiral and a cubicle shaped phenomenon. The results show that our edge detection technique is capable of determining accurate perimeters for not only regularly-shaped phenomena, but also irregularly-shaped phenomena.

V. CONCLUSION

We have presented a robust, distributed, implementable edge detection technique that identifies connected perimeters for sensed phenomena within WSN, without any unit disk communication graph assumptions. Our approach is able to identify connected perimeters of convex shaped phenomena, irregularly shaped phenomena and phenomena that contain connectivity holes. We showed that our technique is robust in topologies of varying densities, average degree and communication range. For future work we will focus on the dynamic adaptation of identified perimeters to maintain accurate geometric representations of changing phenomena in WSN.

REFERENCES