Spatial Processing of Regions in Dynamic Wireless Sensor Networks

PhD Confirmation Report

SUPERVISOR

Dr. Matt Duckham

CO-SUPERVISORS
Dr. Gary Hunter
Dr. Stephan Winter

8-February-2007

Research Scholar:

Muhammad Jafar Sadeq
Department of Geomatics
The University of Melbourne
Email: m.sadeq@pgrad.unimelb.edu.au

Contents

1 Introduction **4**

1.1 Overview 4

1.2 In-Network Information Processing in WSNs 4

1.3 Mobility and Volatility 5

1.4 Motivation 6

1.5 Report outline 7

2 Literature Review **9**

2.1 Introduction 9

2.2 Energy Conservation 9

2.3 Spatial Characteristics of Regions 11

2.3.1 Metric Characteristics 11

2.3.2 Topological Properties 11

2.3.3 Spatiotemporal Characteristics 12

2.4 Boundary Detection 12

2.5 Neighborhood 13

2.6 Routing 18

3 Research Problem **20**

3.1 Research Question and Hypothesis 20

3.2 Research Objectives 20

3.2.1 Static Boundary Processing 21

3.2.2 Boundaries of Dynamic Fields 21

3.2.3 Sensor Mobility and Volatility 21

3.2.4 Inaccuracies Due to Node Location 22

3.2.5 Boundary Vagueness 22
3.2.6 Boundary Anomalies 22
3.2.7 Determine Spatial and Spatiotemporal Characteristics 23

4 Research design **25**

4.1 Methodology 25
4.2 Analysis 25
4.3 Simulation Tools 26
4.4 Implementation on a WSN 26
4.5 Research Outcomes 27
4.6 Thesis Chapter Outline 27

5 Research progress **28**

5.1 Simulation Tool Selection 28
5.1.1 Ptolemy 28
5.1.2 AlgoSenSim 28
5.1.3 Network Simulator (NS-2) 29
5.1.4 GloMoSim and QualNet 29
5.1.5 TinyOS Simulator (TOSSIM) 29
5.1.6 JiST/SWANS 29
5.1.7 EmStar 30
5.1.8 Sensor, Environment and Network Simulator (SENS) 30
5.1.9 J-Sim 30
5.1.10 Repast 30
5.1.11 Conclusion 31
5.2 Planned Algorithms 31
5.2.1 Neighborhood Structure 31

5.2.2 Boundary Tracking . . . 32
5.2.3 Representative Node Selection . . . 33
5.2.4 Sensor Mobility and Volatility . . . 33
5.2.5 Inaccuracies Due to Node Location . . . 34
5.2.6 Boundary Vagueness . . . 34
5.2.7 Boundary Anomalies . . . 34
5.3 Work Completed . . . 34

6 Research Schedule **36**

1 Introduction

1.1 Overview

Technology is advancing to allow in the near future the deployment of a large number of microsensors to detect events in their environment. In many applications it is infeasible to set up a wired network among the sensors and between the sensor network and a controlling station. For example, there cannot be a wired network between the agents in a vehicle ad-hoc network (VANET). In a military zone, there would not be sufficient time to deploy a wired network. In an environmentally remote environment (such as in the ocean), a wired network may be difficult to set up due to either environmental concerns or accessibility problems [21]. Thus, extensive research is being done on *untethered* sensor networks that use wireless communications to achieve connectivity among the sensors and to the outside world.

Wireless communications consume the greatest energy from a sensor's energy reserves. In the VANET application, the wireless agents can draw this energy from the vehicle's engine. However, in the other examples presented, the sensors would need to rely on a battery for their energy needs. A critical focus of wireless sensor network (WSN) research is to lengthen the WSN lifetime by reducing energy loss from wireless communications. Part of such research involves efficient routing in the WSN. Another part involves processing this information so that there is less data to communicate, and fewer nodes with a desire to communicate. This research addresses the second goal by studying ways to efficiently determine spatial characteristics of changing fields in dynamic WSNs using distributed/in-network processing in the WSN. In-network processing can then be applied to VANETs and other non-energy-constrained wireless networks.

1.2 In-Network Information Processing in WSNs

WSNs may be divided into two broad categories based on what the WSN is monitoring — individual objects or fields. Individual objects being tracked may consist of people, vehicles or animals. In contrast, fields generally represent environmental variables such as temperature, humidity and salinity. Sensors detect these environmental variables as point readings, which supports a field-based representation of these variables. This research focuses on WSNs that monitor fields of such variables, and not on monitoring objects.

One of the expected tasks of the WSN is to return some spatial characteristics of a region of a field or the spatial relationships between regions. The simplest manner in which this may be achieved is for each sensor in the WSN to report its reading to a central computer that performs this task. In situations where there are thousands of nodes, each generating readings at a high temporal frequency, reporting a piece of data from each sensor would be a very communication-intensive process that would shorten the WSN lifetime. Practical experiments acknowledge this limitation and implement sleep times to reduce energy usage in order to lengthen sensor lifetime [21]. Additionally, at high densities and frequencies, each individual data item becomes near insignificant. Information processing in the WSN would reduce the data being communicated to improve significance and conserve energy. This would reduce the precision of the data (and may even be qualitative responses), but many applications do not need extremely precise information. Indeed, many applications (such as chemical plant monitoring and control [36]) use quantitative data to generate qualitative results. There is some trade-off between precision and energy usage, but in-network processing of data is expected to provide gains in both energy conservation and relevance of the information.

Among the information processing functions of a sensor node will be to discard insignificant or superfluous data. One example is the growth or movement of regions. Since physical phenomena generally change quite slowly in comparison to the detection, processing and communication ability of the WSN, it is possible that such changes will create a flurry of activity and communication from the WSN. It will be the responsibility of the individual nodes and the WSN to determine how much of these changes to report. There is therefore an additional layer of information processing in dynamic scenarios.

1.3 Mobility and Volatility

In some applications, such as in the ocean, sensors in WSNs may be mobile, which adds a layer of complexity to in-network data processing. In addition, sensors may be volatile — turning off or turning on depending on various factors. Volatility may be an important tool in energy conservation, where sensors turn themselves off during periods when the environment is stagnant. These issues have in general been avoided by the WSN research community, although it is a significant consideration in many WSN applications. This research will consider static nodes, mobile nodes, volatile nodes and combinations of these node behaviors.

1.4 Motivation

There are many applications where preventive action may be taken to prevent disasters, if only the authorities could be aware of the impending crisis. For example, bush and forest fires may be preempted if there were some way to determine where a fire is likely to break out. Similarly, natural preserves may have periods when they are vulnerable to human encroachment. If the authorities could determine when such conditions exist, they could control human activity at such locations. In all such cases, scientists would want to study the environments in detail to understand the causes and effects of various natural (and perhaps mundane) phenomena.

One example that may be studied in detail is the Great Barrier Reef, where a wireless sensor network may be placed to observe and study it. Researchers and scientists are interested in the minute details of the reef's environment, which includes its temperature and the ocean current. Their concern, however, is to detect anomalies in the environment. In contrast, the normal behavior of the environment is less interesting. Managers of the reef are concerned with the larger picture of the reef's environment so that they may control human activity at the reef. These different groups of people have some common properties. Firstly, they are both monitoring environmental fields. Secondly, they are not interested in all the data from the sensors, but only a small subset or summary. In order to satisfy their needs, they may set up the WSN to report only when interesting events occur. For the WSN to determine if an event is interesting, it would need to perform some in-network processing, either qualitative or quantitative.

Qualitative and quantitative information each have benefits over the other. A feature of quantitative data is that qualitative information can be derived from it. In contrast, qualitative data is more meaningful to the users. For example, the scientist may be concerned when a current goes in the opposite direction to a major ocean current (a qualitative property) and the actual direction of either current (a quantitative property) may be unimportant. Additionally, due to the imprecision of sensors in a WSN, quantitative data may be less accurate than qualitative data, which depend on relative values rather than exact values. For example, the statement "The temperature is 25C." may be wrong, but the statement "The region's temperature is normal." is more likely to be correct. For these reasons, the research will concentrate on qualitative spatial reasoning, while not completely neglecting quantitative properties of the environment.

Additionally, in-network information processing in WSNs conserves the sensors' energy. In energy-constrained networks, it reduces the amount of data that is to be communicated, which reduces energy consumption. This will lengthen a WSN's lifetime. It would be very much simpler to collect all the data at a central computer before performing any computation, instead of designing distributed algorithms for the task. However, where energy is a concern, transmitting all the data to a base station is very costly in terms of energy. And even if energy were not a limitation, a central computer would be a bottleneck in the system. The central computer for processing is likely to prove unscalabe as the number of nodes increases.

In-network processing is also useful in WSNs with high node densities and sampling frequencies, where each individual sensor reading becomes less significant. The sensors would have to summarize these readings into compact and meaningful information.

Therefore, the purpose of this research is to develop techniques to perform in-network processing as a tool to conserve energy and to provide salient information from WSNs. The research will focus on determining qualitative spatial and spatiotemporal characteristics of regions of environmental fields.

Among the spatial properties of a region, emphasis will be given to the boundary of a region. Boundary is a defining property of regions, observable and analyzable by human spatial cognition and providing much of the information that defines regions. The boundary can be used to approximate or determine quantitative spatial characteristics of fields such as area, perimeter, width and distance, and qualitative characteristics such as connectivity. Therefore, this research aims to determine the spatial characteristics with a focus on boundary determination as its basis.

There are many possible spatial characteristics of regions, in up to three dimensions, and many temporal processes too. This research begins a solution into the two-dimensional space to enable future work in the third dimension. Also, rather than attempting to cover all spatial characteristics and processes, the research will try to cover the ones that commonly appear in literature as a representative study.

1.5 Report outline

This is a report on the plan to develop energy-efficient and accurate algorithms for in-network processing of spatial characteristics of regions in dynamic WSNs. Boundary will be the basis of the study because it is a fundamental property of regions and many spatial properties can be derived from it. The report begins with a short discussion of this problem of WSNs. Section 2 reviews the existing literature on this issue. Section 3 specifies the research objectives to give identifiable goals. Section 4 discusses the research methodology and the thesis outline. Section 5 reports on the progress. The research schedule is given in Section 6.

2 Literature Review

2.1 Introduction

The main motivation behind the majority of wireless sensor network research is energy conservation. This literature review therefore starts by exploring the problem of energy in WSNs. The review then moves on to the focus of the proposed research — the energy-efficient determination of spatial characteristics of regions in a WSN. Firstly, some of the spatial characteristics that will be studied are discussed. Then, special consideration is given to boundary detection, because that is the expected starting point for determining these spatial characteristics. Fundamental to the definition of boundary is the concept of neighborhood, so literature on neighborhood structures in WSNs are then discussed. The final section of the review deals with communication among nodes in a WSN.

2.2 Energy Conservation

There is no disagreement in the WSN research community that wireless communications consume the greatest energy among all tasks that a sensor node performs. In fact, the transceiver electronics (including startup and listening) consume far larger amounts of energy than actually transmitting the data [35]. Since idle listening is such a significant drain in the sensors energy reserves, it has been suggested that the sensor's radio is turned off at times of environmental inactivity [39] to prevent this energy drain. This is an important technique in energy conservation, and as long as it is a feasible strategy, WSN researchers should consider the effects of such node volatility in their research, and if possible, incorporate it in their investigations.

Additionally, energy models should account for the various parts of the node's activity that consume energy [23]. Processing certainly consumes less energy than wireless communication, but there is also certainly a point where the information processing energy requirements overtake the energy conserved by reducing some communication. Having an energy model that incorporates all the possible losses of energy would enable the researcher to observe the gains of each step of his/her work.

In some applications, it would be possible to develop nodes that can harvest energy from the environment, using technologies such as solar panels. Not all nodes will have access to solar energy. Those that do can be made mobile to obtain the energy from sunlit locations and return to divulge the energy to the network. However, the experiments done in this regard are estimation calculations using known energy costs rather than simulations with the devices [29]. Solar energy, if uninterrupted, can enable self-sustaining WSNs. It can provide up to 15mW/cm^2 given direct sunlight, and 0.15mW/cm^2 on a cloudy day [28]. In comparison, the CC2420 transceiver uses 4.8mW for 8kbps reception only [6], which is already more than that available on a cloudy day. When combined with other energy requirements such as transmission, computation and idle listening, and the fact that direct sunlight is available for only a small portion of the day, it can be seen that harvesting solar energy does not provide sufficient energy for a sustainable WSN.

Another energy harvesting technique is using piezoelectronics to gain energy from vibrations, which is especially useful in vibrating structures such as bridges. In this technique, it is necessary to measure the vibrations of the structure being monitored and design the energy harvester to work at the measured frequencies, and there should be sufficient power in the vibration [1]. The authors state that this would enable "battery powered systems to last longer", implying that such energy-harvesting techniques only serve to prolong a WSN's lifetime. There is in fact a large gap between the harvestable piezoelectric energy (0.1mW/cm^3 [28]) and the energy required for 8kbps wireless reception (4.8mW [6]), neglecting other sensor tasks.

As such, in-network processing of spatial characteristics can aid in conserving energy to enable self-sustaining WSNs, and also to increase WSN lifetime in applications where energy scavenging is not a possibility, such as an underground location, which would in general neither have access to sunlight for solar panels nor have sufficient vibrations to enable piezoelectric harvesting.

There are some wireless networks that have comparatively limitless energy resources. One such application is the vehicle ad-hoc network (VANET) [14]. In environmental monitoring, the information consumers are relatively far from the WSN. In contrast, the information consumers of a VANET are usually the vehicles themselves. To avoid the bottleneck that a centralized information processor might create, the nodes on the vehicles should be able to process the information among themselves. This example illustrates that in-network processing may be relevant to non-energy-constrained wireless networks. Some of the tools or algorithms that are designed in this research may be usable in such wireless networks that require in-network processing.

2.3 Spatial Characteristics of Regions

The proposed research will involve attempting to discover energy-efficient techniques for computing spatial characteristics and temporal processes of regions. They fall under the following broad categories.

2.3.1 Metric Characteristics

Metric characteristics are quantitative aspects of the regions being measured. Metric characteristics of regions include the coordinate location of the region, area, perimeter and centroid [37]. There are also metric characteristics between two regions, such as distance. Standard algorithms for these are available in literature [37]. The task is to adjust the algorithms to the domain of WSNs. As an example of the computation of metric characteristics, the WSN may have to monitor the area of a hot region.

The in-network computation of the area of a region has received special attention from some researchers in WSN. They have suggested a simple inverse neighborhood method to compute area in a distributed manner [8]. Each sensor computes its representative area using the node density of its neighborhood. The total area of a region is the sum of all areas of sensors in that region. The authors assume a tree is created by the nodes in a region when the region is detected, and that a node is selected arbitrarily to be the root. This method of computing the area requires communication among all nodes in the region.

In comparison, a boundary method should need only communication among the boundary nodes, and further processing of the boundary will yield the area. This research will thus be providing a new algorithm for area determination in WSN. This method is expected to be less costly in terms of energy usage than the inverse neighborhood method.

2.3.2 Topological Properties

A topological property is a property that is invariant under homeomorphisms. A homeomorphism is a continuous bijection (a one-to-one and onto mapping betweens sets) between two figures whose inverse is also continuous [37, pp. 99, 107]. Examples of topological properties are connectivity [37, pp. 109–113], the topological relationship between points and regions [37, pp. 198–200], and the topological relationships between regions

[7]. For example, the WSN may have to report whether there are holes in an oil slick, or whether there is a large region of stagnant water in a hot region.

2.3.3 Spatiotemporal Characteristics

Temporal characteristics represent the dynamic aspects of the region. Some temporal aspects describe events and processes occurring in the region, such as whether the region is splitting, moving or growing. This has received attention in one research work [38], using the set operation *subset* to determine whether a region has grown/shrunk or moved. Spatiotemporal properties can described using metrics such as velocity of the region's centroid and its growth rate. In the Great Barrier Reef example, the WSN may be required to generate an alert if a hot region is growing at a fast rate.

Regions can be defined by a simple property (such as high temperature), or they can be based on set-based operations on multiple regions. For example, users may be interested in the *interesection* of regions of high temperature and low humidity in a forest. The regions resulting from such set-based operations can also be analyzed for their topological, metric and spatiotemporal characteristics.

2.4 Boundary Detection

The plan in this research is to determine the majority of the spatial and spatiotemporal characteristics of regions from the boundaries of the regions. Thus, the first step would be to determine which nodes constitute a region's boundary.

Some of the existing research on boundary in wireless sensor networks is concerned solely with determining whether a node is a boundary node. One such research proposes a classifier approach where each node collects information about its neighborhood. It then computes a line that partitions its neighborhood into nodes that are in a region and those that are not. Finally it compares its distance from that line to a threshold to see whether it is on the boundary [4]. In many situations, a region may be a range of values, rather than a single value. In such a case, the boundary node can be considered to be any node where the average reading of its neighborhood is close to a threshold value [12]. Averaging also reduces noise in the sensor readings. These works do not continue on to process the complete boundary of the region.

Another approach [26] involves hierarchically partitioning the WSN. Each parent node estimates a single reading for its child nodes and transmits this up the tree. Their approach is optimal when they try to estimate the boundary by processing the entire spatial region of the WSN. It is expected that a system can be designed that involves only relevant sensors. This would be more efficient because those sensors not on the boundary will not need to communicate their readings, saving their energy resources.

One approach uses a need-based triangulation of the WSN [5]. A node is a boundary node if it has detected a phenomenon and its neighbors in the triangulation have not detected it. The triangulation adjusts itself when nodes turn on or turn off, so this is a unique work that compensates for volatility in its algorithm. This volatility is in response to the dynamism in the field being monitored, thus it also provides for field dynamism. In contrast to this triangulation, there exist triangulations (such as the Delaunay triangulation) that closely represent the neighborhood of the nodes. Therefore, it is expected that such a triangulation will not give the most accurate results.

In a continuation of this line of research, the triangulations are then used to set up transition rules for changing regions [38]. If a phenomenon is detected inside a certain triangle where it was not detected before, that triangle's relationship with its neighbors can be used to determine the connectivity of the region and whether the region is splitting. However, this relies on incremental changes to region, and do not always work when two triangles change simultaneously. There are also many other interesting spatial characteristics that can be studied. These works provided the initial ideas that led to this research proposal, and some of the ideas presented therein may prove to be usable.

All the previous works are lacking in many areas — none of them consider node mobility and only one line of research considers volatility and dynamic regions. This research plans to build on this existing work in its boundary detection and tracking algorithms, and to use the boundary to compute various characteristics of the field(s) being monitored. The first stage in such boundary-based algorithms is first determining the neighborhood of a node.

2.5 Neighborhood

Boundary is formally defined using the concept of neighborhood. In a set of points S, the neighborhood of a point x is defined as a subset of S that includes an open set containing x. In simpler terms, the neighborhood of a point is a set containing the point where the point can be moved slightly without leaving the set.[1] Boundary is then defined as follows. Let S be a topological space (a collection of neighborhoods). Let X be a subset of points in S. An individual point x in S is *near* X if every neighborhood of x contains some point of X. The boundary of X consists of all points that are near to both X and X' [37].

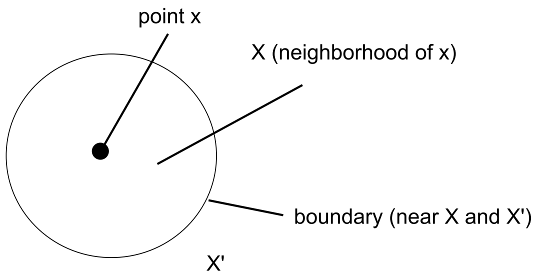

Figure 1: Neighborhood and Boundary

Therefore, boundary requires as its basis the existence of some definition of neighborhood. Thus, one of the aims of the research is to study existing neighborhood structures in WSNs and to suggest one or more new structures, and then to compare these structures.

Most of the existing WSN research that is specifically related to boundary detection (Section 2.4) use the simple definition of neighborhood where all nodes in a node's communication range consist of its neighborhood. If the node density is high, it may lead to the existence of many nodes that may be construed as the boundary nodes. Fig. 2 illustrates this. Node A considers itself a boundary node because it communicates directly with Node B on the other side of the boundary. However, there are many nodes closer to the boundary than Node A. A refined boundary will not include Node A due to there being better choices of nodes as boundary nodes.

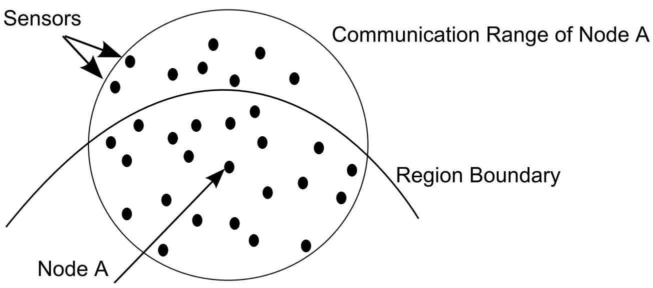

Figure 2: Boundary using Communication Range as a Neighborhood

From Section 2.4, both the averaging method [12] and the classifier approach [4] use threshold values to reduce the probability that nodes like Node A are boundary nodes.

[1] Wikipedia, `http://en.wikipedia.org/wiki/Neighborhood_(topology)`.

Since sensor nodes are unlikely to be located exactly on the boundary, any boundary determined from the WSN is only an approximation to the real boundary. A good neighborhood structure will reduce the error in approximating the boundary. Thus one of the tasks of this research is to find a neighborhood structure that optimizes the error in boundary determination.

Many neighborhood structures are based on triangulations. A triangulation of a set of points has the maximum number of edges possible in the graph without any of the edges intersecting [27]. Triangulations thus discretize space into the largest number of units given a fixed set of vertices. In other words, a triangulation gives the highest possible resolution of space, which leads to the lowest possible error.

One triangulation is the Delaunay triangulation (DT). For a set S of points in the Euclidean plane, the Delaunay triangulation $DT(S)$ is the unique triangulation of S such that no point in S is inside the circumcircle of any triangle in $DT(S)$ [27].

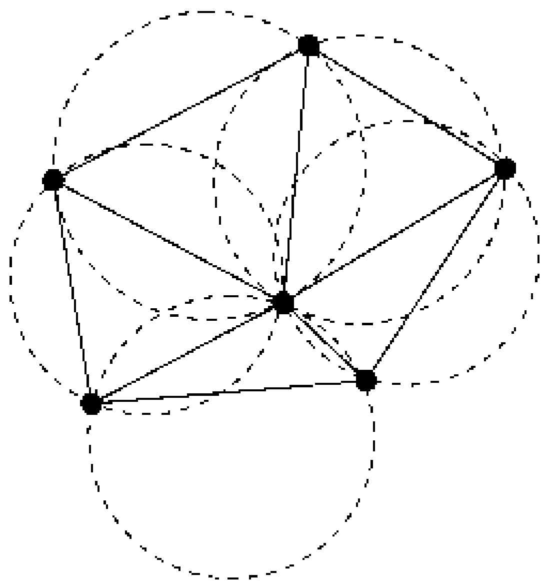

Figure 3: The Delaunay Triangulation [37]

There has already been some research on DTs in WSNs [19]. It is shown that a node's DT using only its 1-hop neighborhood may have edges that cross those of another node's DT. This occurs when a circumcircle is larger than the communication range of a node. In such a case, a node may be unaware of a node just outside its communication range that invalidates the DTs that it has computed. In Fig. 4, node u forms a DT with v and w. Node u cannot communicate directly with node y, which lies inside the circumcircle of Δuvw and invalidates it as a Delaunay triangle. Furthermore, the triangle Δxyz has edge xy that intersects edge uv of Δuvw, resulting in a non-planar structure.

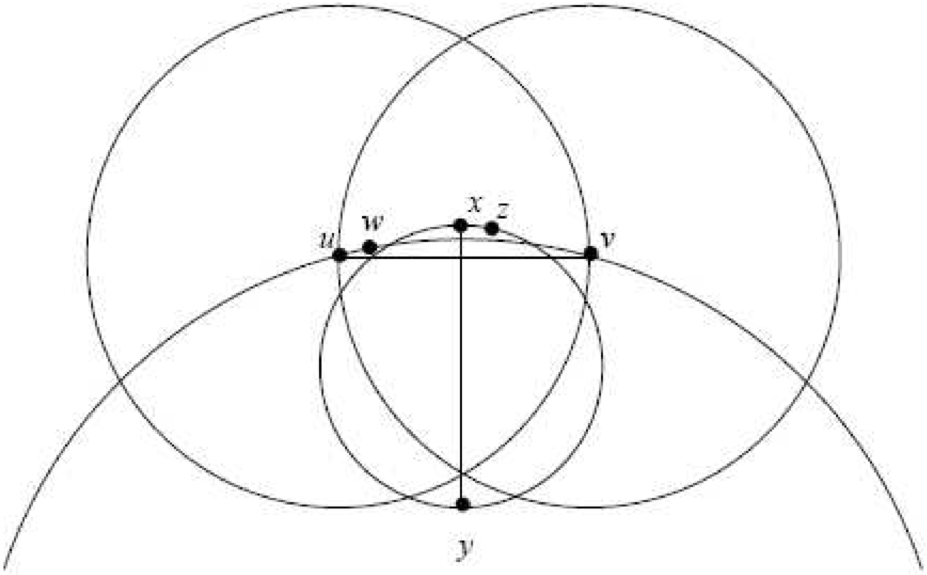

Figure 4: Local DT using 1-hop Neighborhood is Non-Planar [19]

If neighbors from a node's 2-hop communication range are used, the resulting structure is planar [19]. However, the 2-hop communication range algorithm that has been presented involves a lot of communication — an announce message, a broadcast of its computation of some triangles, acceptance messages for each of its neighbors' triangles that it wishes to confirm, a check message, and a final confirmation message. At the end of this step, a supergraph of the 2-hop Delaunay triangulation is obtained. This is a large amount of communication and coordination that cannot be justified in a situation where the nodes are highly mobile or volatile. Furthermore, it may not even be necessary, since a non-planar structure is likely to be suitable for the boundary detection task.

Therefore, one of the structures that will be investigated is the approximate DT using information from nodes in a 1-hop communication range. This will significantly reduce the communication requirements compared to a 2-hop system. It would however introduce the possibility of a non-planar graph. Planarization is an important consideration in routing to prevent loops, but whether this causes any critical problems for boundary detection will need to be investigated.

It should be noted that the sparser the WSN, the more likely it will be that the true DT involves nodes that are not in direct communication with each other. In such cases, the localized DTs will show greater deviations from the true DTs (Fig. 4). This is why

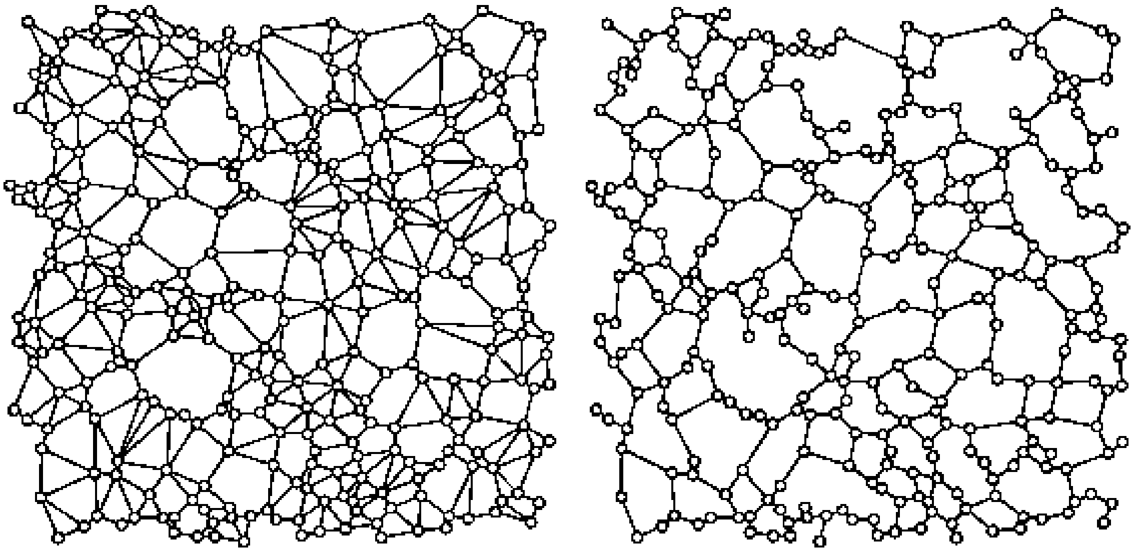

Figure 5: The Gabriel Graph (left) and the Relative Neighborhood Graph (right) [41, pg 79]

researchers attempting to perform local computations of DTs for sensor networks call their structures restricted Delaunay graphs or local Delaunay triangulations [19].

Other neighborhood structures include the Gabriel Graph (GG) and the Relative Neighborhood Graph (RNG) [27]. These structures are sparser than the DT, and in fact are the subgraphs of the DT. They thus have fewer edges than the DT. This creates a low resolution discretization of space, with the associated errors of lower resolution representations of information. Fig. 5 shows an example of the GG and the RNG. It can be seen that there are regions in both graphs where boundary tracking will be difficult due to the lack of edges.

Other triangulations include the greedy triangulation [37], which attempts to minimize the total edge length in the triangulation.

It is also possible to derive need-based triangulations. An algorithm has been developed to do so in WSNs [5]. This algorithm involves communication among the neighbors to ensure a planar triangulation structure. It incorporates in its basic structure the possibility of node volatility, although node mobility may be a concern that will need to be addressed in extensions to the algorithm. Although a need-based triangulation is not expected to be as good as a Delaunay or greedy triangulation, it is still worthy of investigation due to the algorithms inherent allowance for node volatility.

It is uncertain whether a triangulation is necessarily the best structure for boundary detection. Nor is it certain that a planar structure is necessary. Thus, the research will not necessarily limit itself to planar triangulations.

The best neighborhood structure will optimize the error in the boundary approximation and the energy required to maintain the structure under dynamic conditions. The error will generally be less with shorter edge lengths because the vertices of a shorter edge are closer to the location of the physical boundary. It is interesting that neither the DT nor the greedy triangulation minimize the total edge length in the neighborhood structure [24]. Instead, there exist minimum weight triangulations (MWTs), having the minimum possible sum of edge lengths. However, computing MWTs is computationally expensive (non-polynomial hard [24]).

Algorithms to *approximate* the MWT are of polynomial order of complexity. The fastest algorithm that has been suggested so far takes quasi-polynomial time ($O(n^{polylog(n)})$) [30]. The conclusion of the researchers who developed the algorithm is that their work gives hope that a polynomial time algorithm can be found in the future.

2.6 Routing

Local communications among nodes are generally broadcast to the neighborhood. However, the determination of some spatial characteristics (such as area) requires communication among many nodes in the region. If there is one node that handles the collection and summarization task, there must be a way to transmit updates to that node. This requires routing in the WSN.

Routing is one of the most studied areas in WSNs. Some of these routing algorithms are applicable to situations where the location of the recipient is known, such as a particular node, a region with a known location, or the base station. These address-centric algorithms may be divided into greedy algorithms [10, 13, 15, 16, 17, 31] and hierarchical architectures [9]. In contrast, data-centric algorithms [3, 11, 2] query the network looking for some information without prior knowledge of the location of this information.

Many address-centric routing protocols are based on greedy routing, where, whenever a node has some information to send to a destination, it checks its neighbors for the node that is closest to the destination and forwards it to that node. There is a possibility that the packet may reach voids where there are no nodes closer to the destination than the current hop. To overcome this, the Greedy Perimeter Stateless Routing (GPSR) [13] algorithm includes a recovery phase where the information is routed along edges of a planar graph until it reaches a point where greedy forwarding may be used again. This recovery procedure is called geographic face routing. Its shortcomings due to location inaccuracies are resolved by other researchers [16, 31]. Geographical and Energy Aware Routing (GEAR) [40] is another greedy strategy where the next hop chosen is *closer* to the destination than itself, rather than the closest. This routing mechanism suits finding paths around voids, and GEAR does not therefore include a completely different recovery algorithm.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [9] groups nodes into clusters with a single cluster head that is the recipient of all data of the cluster. There may be a hierarchy of cluster heads so that any information is passed higher up the tree. The top cluster head transmits directly back to the base station. One major contribution of this work is the demonstration that if a long range communication is broken down into hops, the nodes nearest the destination will quickly be depleted since all communication passes through them. Thus LEACH distributes the workload by rotating the role of cluster head among the nodes.

Among the data-centric algorithms is directed diffusion [11], which initially uses flooding to discover the destination. It is possible that this discovery may be done by controlled flooding [3]. Once the target is discovered, directed diffusion creates several paths between the query source and the information source for robustness in case of failure of some of the paths. Another data-centric algorithm, rumor routing [2], makes a trade-off between the high success rate of directed diffusion and the low energy usage of single-path communications. Whenever there is an event or a query, an agent is created by the source. This agent hops from one node to another at random, recording its information at each hop. When a query agent crosses the path of an event agent, or vice versa, it follows the path made by that agent to the source.

Boundary tracking does not directly involve routing, being an application layer problem. It will be assumed that there is a network layer mechanism to transmit information from the region of interest to the base station, and for queries to reach the region of interest. Boundary tracking will involve 1-hop neighbor communication, which is not a routing problem. However, it will also involve communication between nodes further than 1-hop (for example, to report updates to a node that represents the region), and for this, the above research on routing is applicable. The proposed research will assume that such a routing structure exists at a network layer. This research will not study the efficiency of various routing algorithms, nor implement any of them.

3 Research Problem

3.1 Research Question and Hypothesis

Some work has already been done on boundary detection, but these do not completely deal with boundary processing and network and field dynamism. There are many spatial characteristics of regions that are non-trivial but have not been dealt with in WSNs. Thus the research question for this proposal is:

> *"How can the properties of a region in a dynamic field be efficiently detected and tracked in static, mobile and volatile wireless sensor networks?"*

The research hypothesis is:

> *"In-network processing of the boundary of a region enables an accurate, efficient and concise computation of many dynamic spatial characteristics."*

Accuracy can be measured by comparing the results of the algorithms to the actual properties of the region. A WSN is not expected to be highly accurate, but the results should approximate the true properties of the region.

An efficient computation of dynamic spatial characteristics would require less energy than other methods of computation. The energy requirements of methods that have been proposed in the literature can be compared to those of the algorithms developed in this research. For spatial characteristics that have not been studied in literature, the efficiency will be compared to some basic means of determining the spatial characteristic (such as querying all the nodes in the region).

Boundary, being fundamental to topology, and also being a salient property of regions, allows the derivation of many spatial characteristics. Instead of having various means of computing these properties, it is hoped that boundary can be used to compute the large majority of spatial and temporal characteristics. If this can be done, the concept of boundary will have been shown to be a concise foundation for such tasks in WSNs.

3.2 Research Objectives

The research will involve the following tasks.

3.2.1 Static Boundary Processing

The literature review has discussed a few methods of boundary detection (Section 2.4). In two of these strategies, the nodes process information in their communication range to calculate their own status as boundary or non-boundary node. These methods are local and involve only the boundary region. In another method, information is hierarchically passed up a tree structure. This method involves all nodes in the network. Another possibility that has been suggested is the use of triangulations. Triangulations will be the basis for boundary detection in this research. One objective of the research is to test the efficiency of each of these strategies with respect to requirements such as energy usage and accuracy.

All of these strategies enable each node to determine whether it is a boundary node of a region. Information about these boundary nodes will then need to be collected. This in-network boundary processing may involve, among other things, passing a message along the boundary, selecting representative nodes to perform the final summarizations, and then actually performing these summarizations.

3.2.2 Boundaries of Dynamic Fields

Fields may change in many ways, each of which needs to be dealt with. Among the ways in which a field may change are growth, shrinkage, splitting, appearance and disappearance of regions. The research will deal with tracking these changes in the boundaries efficiently. This will involve decisions on what changes to report and what changes to ignore, and how to efficiently report the significant changes.

3.2.3 Sensor Mobility and Volatility

The research will deal with sensors moving in and out of fields, which changes the node composition of the boundary and also affects the neighborhood structure. In constantly moving environments such as in bodies of water, the research would be to ensure that the sensors minimize the processing and communication that occurs due to these changes and only deal with significant mobility.

The node density of the boundary may also vary due to nodes awakening from sleep mode and also nodes turning off, which may occur even during periods of high environmental activity if a node wishes to conserve its energy for future energy scavenging operations.

Again, these changes in boundary node density will be dealt with so that only significant changes are reported.

3.2.4 Inaccuracies Due to Node Location

The research will analyze the adverse effects of node location inaccuracies in both the neighborhood structure and the boundary processing algorithms. Since boundary is based on neighborhood, and the neighborhood structures being studied are based on location information, location errors may give rise to significant errors in the characteristics being tracked.

There are many possible reasons for node location inaccuracies. One common reason is that sensors are placed in a three-dimensional world, but their coordinate locations are projected onto a two-dimensional plane. Thus, distances between nodes will in general be greater than the expected distance that the nodes calculate. Other errors include random errors in all nodes and significant errors in a small subset of nodes.

3.2.5 Boundary Vagueness

A vague boundary is one where the exact location or line of the region boundary is indeterminate. For example, if the definition of the region is qualitative (a "hot" region), there will often be some regions between the hot and cold regions that can be considered to be neither hot nor cold. This region has a vague boundary. Boundary nodes (or the representative node) should be able to determine whether the boundary is vague or crisp depending on the definitions of regions given to them by their users.

3.2.6 Boundary Anomalies

Boundaries may have other complications. One such scenario is a front, where the region that is being monitored is not fully enclosed in the WSN. This is a significant issue because nodes on the perimeter of the WSN may or may not consider themselves boundary nodes. Other anomalies include fields that have sections that are only one-node thick and fields whose boundaries pass through holes in the WSN.

3.2.7 Determine Spatial and Spatiotemporal Characteristics

All the previous objectives deal with detecting and tracking the boundary. However, the goal of the research is not to simply detect the boundary, but to also use it in computing characteristics of the region. These characteristics, and some ways they may be evaluated, have been discussed in Section 2.3.

For each spatial property, there may be more than one method of determining it. For example, Fig. 6 shows two ways in which the intersection of two regions may be found. Fig. 6(a) demonstrates the result from the standard computational geometry algorithm for determining intersection [37, pg. 202]. However, due to implementation or other concerns, WSNs may include only the sensor nodes in the vertices of the intersection region. Then, the result would be Fig. 6(b). This example demonstrates just one of the issues that are to be addressed in the determination of spatial and spatiotemporal characteristics.

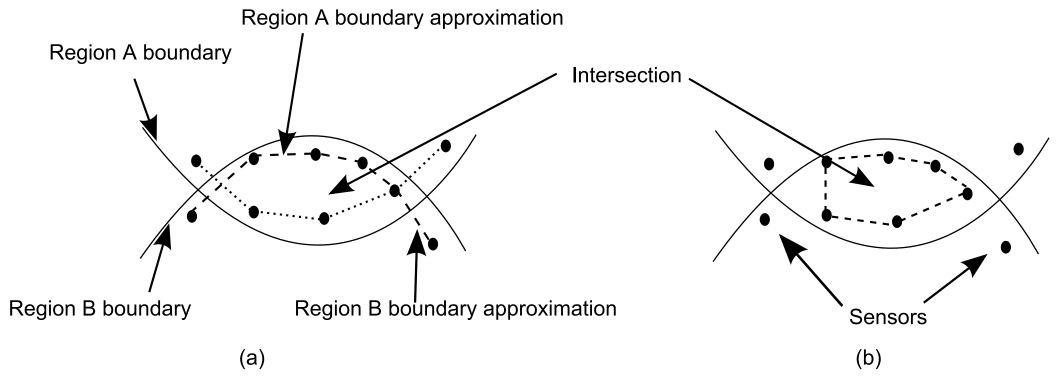

Figure 6: The Set Operation 'Intersection' in a WSN. (a) Two regions and their approximate boundary, (b) Intersection using standard algorithms, (c) Intersection directly from sensed values

To summarize, the research will involve the following.

1. Investigate existing work on static boundary detection, tracking and summarization, and suggest new techniques for these tasks.
2. Investigate the suitability of these techniques tracking dynamic fields and mobile and volatile sensor nodes.
3. Deal with inaccuracies in node location
4. Deal with vague boundaries

5. Deal with anomalies in regions.

6. Compute other spatial characteristics given the processed boundaries.

4 Research design

4.1 Methodology

The research will consist of developing structures and algorithms for the tasks enumerated in Section 3.2. These algorithms will be analyzed for their effectiveness through simulation of a network of sensor nodes. The model of these sensor nodes will be represented analytically. One tool being studied for this representation is I/O automata.

I/O automata provide a concise means of representing the functionality of processes. Processes are represented as a set of actions (input, internal and ouput), states of the process, and transitions that define how the actions affect the states [20]. I/O automaton representation of processes can lead to proofs such as safety (ensuring that nothing bad happens [20]). However, the basic I/O automata model is only useful for static scenarios. An extension called Dynamic I/O Automata may be suitable for this research.

4.2 Analysis

The neighborhood structures and algorithms need to be tested for their accuracy. This will be done through simulation, and, where possible, analytically. In addition, there may be the possibility of performing tests on a real sensor network. The tests done using simulation will involve measuring one or more performance metric for each test being made.

The various neighborhood structures will be tested based on their accuracy in approximating the real boundary, given different setups of nodes and regions. This will be done by computing the set-oriented difference between the real area of the region and the area that results from the approximate boundary. The neighborhood structure that gives the lowest difference consistently would be the best. It may be that different neighborhood structures are better under different conditions, such as density of the sensor network. The algorithms to track the boundary will be analyzed in the same manner.

Other accuracy measures for boundary may include accuracy of the qualitative shape of the region, rather than the area.

Algorithms to deal with dynamism will involve cost-benefit analysis, with the cost being the communication required (which depends on the frequency of updates), and the benefit being the average accuracy (over time) of the boundary approximation, measured as explained above. The better algorithms will use less updates to give higher accuracy.

4.3 Simulation Tools

A simulation tool will be selected that provides:

1. the ability to simulate dynamic fields, since this research involves processing of field boundaries, and also to create vague regions within fields.
2. the ability to define and visualize various neighborhood structures in the network.
3. the flexibility in determining node locations and distributions, and in making the nodes mobile and/or volatile.
4. scalability (the ability to simulate thousands of nodes, or more, in a reasonable time).
5. the ability to represent energy usage and energy reserves, so that these could be analyzed for comparison between different structures and algorithms.
6. the ability to model uncertainty such as location inaccuracies.

If a simulation tool is not found that meets these requirements, one would need to be built. A complete simulation tool would also include datalink and network layer communication protocols and signal loss models. However, the lower-level protocols are outside the scope of this research and signal loss will be ignored as a different problem to the one being studied.

4.4 Implementation on a WSN

Given the availability of sensors, the algorithms that are developed may be implemented on a real sensor network. This would enhance the studies in this research. However, it is as yet uncertain whether real sensors will be used during the research period.

4.5 Research Outcomes

The research intends to produce an algorithm or a family of algorithms (with systematic comparisons between algorithms) for processing the boundary of regions in WSNs and thence suggesting how other spatial characteristics may be computed.

4.6 Thesis Chapter Outline

The thesis will be structured into five chapters.

- Chapter 1: Introduction
  - Motivation behind the research
  - Identification of goals and objectives
  - Outline of the thesis
- Chapter 2: Literature Review
  - Energy limitations of wireless sensor networks
  - In-network processing
  - Boundary in wireless sensor networks
  - Neighborhood structures
  - Routing
- Chapter 3: Methodology
  - Algorithms for neighborhood structures
  - Algorithms for calculating spatial and spatiotemporal characteristics
- Chapter 4: Evaluation
  - Simulation tool selection
  - Analysis of algorithms
  - Discussion
- Chapter 6: Summary and conclusions
  - Discussion and conclusions
  - Future directions

5 Research progress

5.1 Simulation Tool Selection

The research cannot progress without a tool to test new ideas. It is not expected that any of the tools will provide all the desired functionalities, but the ease of extension to include these functionalities is a major criterion. The simulation tools that have been analyzed so far are discussed below.

5.1.1 Ptolemy

Ptolemy (`http://ptolemy.eecs.berkeley.edu/`) is a simulation tool that has a very useful graphical representation of objects and allows the building of objects with customized functionalities. It allows the simulation of application level problems without requiring the user to address or deal with lower layer functions such as medium access and routing. It allows the creation of mobile sensors.

However, due to its object-oriented nature, scalability is a concern. Programming would be required to include in the nodes memories for links, structures, recent information and to collect information for processing, or these would have to be handled in some other way. Another issue is that it only simulates circular regions. Regions of irregular and varying shape would require additional work.

5.1.2 AlgoSenSim

AlgoSenSim (`http://tcs.unige.ch/doku.php/code/algosensim/overview`) is a new sensor network simulator (it's currently version 0.9). It is designed to simulate distributed algorithms, which is the focus of this research. It permits dynamically changing the number of sensors (volatility) and moving the sensors. It can also generate events to trigger the sensors. It includes a graphical user interface to view connections. It has inbuilt support for localization errors. Unfortunately, Algosensim could not be downloaded for further study because of a broken link.

5.1.3 Network Simulator (NS-2)

NS-2 (`http://nsnam.isi.edu/nsnam/index.php/Main_Page`) is an object-oriented simulation tool, and thus comes with a scalability problem. It supports the simulation of limited power. An extension permits triggers external to the WSN. One major drawback is that NS-2 lacks an application model. It is thus suitable for testing routing and MAC protocols, but application layer algorithms would be difficult to implement.

SensorSim (`http://nesl.ee.ucla.edu/projects/sensorsim/`), is an extension designed for wireless sensor network, but has been removed from the website and is no longer supported.

5.1.4 GloMoSim and QualNet

GloMoSim (`http://pcl.cs.ucla.edu/projects/glomosim/`) is another popular WSN simulation tool. It is effective for simulating IP networks, but cannot accurately simulate sensor networks. It does not support external phenomena triggering the network. It gives extensive freedom in node placement. GloMoSim is no longer being updated. The commercial version, QualNet, supports mobile nodes, but mostly extends GloMoSim to networks other than mobile ad-hoc networks. QualNet does not significantly increase the functionalities for simulating WSNs over GloMoSim.

5.1.5 TinyOS Simulator (TOSSIM)

TOSSIM (`http://www.cs.berkeley.edu/~pal/research/tossim.html`) is a WSN emulator that enables the simulation of actual WSN code in the TinyOS system and language. TOSSIM is highly scalable. TOSSIM simulates the data gathering capability of sensor nodes, but does not simulate the external phenomena that should trigger these events.

5.1.6 JiST/SWANS

JiST/SWANS (`http://jist.ece.cornell.edu/`) is a Java-based simulator that is more scalable than NS-2. SWANS is the sensor network extension to JiST, but the documentation for it is very weak. It has some similar features to GloMoSim, but the lack of documentation hinders a deeper analysis. Using JiST alone would involve significant programming, but that also means that it would be highly flexible. If JiST were the chosen simulation environment, SWANS would be given a more involved analysis to determine its suitability.

5.1.7 EmStar

EmStar (`http://cvs.cens.ucla.edu/emstar/`) is an emulator that runs on the Linux platform. The documentation for EmStar does not actually discuss what EmStar can be used for, but instead discusses programming in EmStar. EmStar is geared towards testing real sensor network code rather than application-level algorithms.

5.1.8 Sensor, Environment and Network Simulator (SENS)

SENS [32] has good capabilities to model environmental factors that affect signal propagation. However, the environment is tile-based and therefore it will be difficult to make the environment grow or shrink to create dynamic fields. It also has a global energy model, rather than a per-node energy model. It also does not provide a means to create neighborhood structures.

5.1.9 J-Sim

J-Sim (`http://www.j-sim.org/`) is a highly scalable simulator with an energy model that supports modeling both the CPU and the radio power consumption. It supports phenomena detection and application-level algorithms may be simulated. It allows per-node in-network processing and includes WSN routing protocols. However, it is still uncertain whether J-Sim could be extended to simulate fields with dynamic regions. However, a graphical view of the sensor network would need to be programmed.

5.1.10 Repast

Repast (Recursive Porous Agent Simulation Toolkit) (`http://repast.sourceforge.net/`) is a Java-based agent simulator [25, 34]. It is under active development and has a medium-sized community for user support. It has an advanced graphical user interface, but at the same time gives large programmatic control over the simulation display. It allows the chaging of parameters dynamically (during a running simulation). It does not impose limitations on the number and complexity of agents, and has efficient memory management.

It supports some random generation of agents based on means and standard deviations, but more complex requirements such as intentionally creating sparse regions would need to be coded. It allows for limited range communication based on distance from agents. Agents can be destroyed and created during the simulation execution, providing support for node volatility. The agents can also move. At the moment, the only minor difficulty seems to be in controlling agent start locations, and that can probably be handled with some programming.

5.1.11 Conclusion

Repast and J-Sim seem to provide many of the features required. Repast provides an easy-to-build display, unlike J-Sim. Ptolemy provides graphical features, but has some scalability issues — tens of nodes could be simulated in a few seconds, but a hundred nodes took half an hour. Repast will therefore be investigated next, and if that is unsuitable, J-Sim would be the next simulation environment. If none of these suits, JiST/SWANS will be used as a basis for programming a simulation system.

5.2 Planned Algorithms

Initial plans have been made for the algorithms to perform some of the tasks discussed in the research objectives (Section 3.2). These plans are discussed below.

5.2.1 Neighborhood Structure

Boundary detection will involve the creation of a neighborhood structure. In order to achieve the energy benefits of local processing, each node will maintain a neighborhood structure that is independent of other nodes' structures. At the minimum, a node should be aware of the nodes in its communication range in order to create this neighborhood structure. This leads us to a few different structures.

Currently, there is only one work in the WSN literature that uses triangulations to detect a small subset of spatial characteristics of regions, and it deals with only incremental changes in the region(s) being monitored [38]. Some of the concepts can be reused from this work to observe the connectivity of regions. However, many spatial characteristics, such as metric characteristics, need new algorithms to be developed, and the assumption of incremental change needs to be relaxed. Some of these newly-designed algorithms can be based on existing algorithms. For example, some spatial characteristics require the boundary information to be collected by one node. There exist election algorithms in the distributed algorithm literature to select this node. Similarly, DT and greedy triangulation algorithms can be drawn from the extensive work available on computational geometry.

The neighborhood structures to be tested are described below, with some description of how they will be implemented.

1. **Unit Disk Graph:** In this basic neighborhood structure, the node does not perform any further processing. All nodes in its communication radius are its neighborhood. It can be seen from Fig. 2 that this is unsuitable for boundary tracking because of a large number of non-boundary nodes incorrectly determining that they are boundary nodes. *The unit disk graph has already been implemented* (Section 5.3).

2. **Delaunay Graph:** The node will use the list of the N nodes in its communication range S to calculate a Delaunay graph of this portion of the WSN. It will sort the list of nodes and recursively split the list into two sections until each section has 2-3 nodes. It will connect these nodes and merge the results of each splitting operation [18].

3. **Greedy Triangulation:** The node will create a greedy triangulation of the N nodes in its communication range S. It will do so by generating a list of all edges between points of S. It will then remove the shortest edge from the list and add it to the triangulation as long the edge does not cross an existing edge. The procedure stops when there are no edges left or the number of edges is equal to 3N-6 [27, pg 228]. *The greedy triangulation has already been implemented* (Section 5.3).

4. **Need-Based Triangulation:** Triangulations are created when needed. Using an existing algorithm for this purpose [5], nodes will communicate to the relevant neighbors their desire to create a triangle. Triangles are only created if all three nodes in the triangle agree to it.

5.2.2 Boundary Tracking

A node will first determine whether it is a boundary node by checking its neighboring nodes for one that is not in its region. It will then broadcast a boundary tracking request, which will be picked up and propagated by nodes in its neighborhood that are boundary nodes. The result will either be returned to the originating node, or it will be informed that another node has taken over the responsibility of tracking the boundary.

5.2.3 Representative Node Selection

Some of the algorithms may require summarizing some properties, and this would require a representative node. It is possible that more than one node attempts to take charge of tracking and summarizing the boundary and other spatial characteristics. If this occurs, it will be inefficient to independently track the boundary for all these requests. There should be an election among the nodes to select one node that will be responsible for boundary tracking.

One election algorithm in distributed algorithm literature is extinction. The extinction algorithm involves a message passing through all the nodes. The message contains the current nominee for the role of the representative node, and the reason why that node is the nominee. As long as a node does not receive a message, it considers itself to be the representative node. If a received nominee is better than itself, it stores that information. If it has a better reason to be a nominee, it replaces the nominee in the message with itself. In either case, it forwards the message to other nodes [33]. Some adjustments need to be made for WSNs, such as limiting the message passing to relevant nodes only, and to allow multiple representative nodes to exist if necessary.

One criterion for selecting a representative node may be the distance from the base station. After the boundary has been tracked, this representative node may transfer its duties to another node for performance reasons. For example, it would be preferable to have a representative node in a rapidly changing region of the field in order to minimize the distance update messages need to travel.

5.2.4 Sensor Mobility and Volatility

If a node moves, it should decide whether the movement is sufficient for it to broadcast its new location. One simple energy-saving idea may be that a node does not care about maintaining an up-to-date neighborhood structure until it is in the neighborhood of a spatial boundary. If there is movement in a neighborhood of interest, the moving node needs to determine whether the movement is large enough to invalidate previous conclusions about the boundary.

5.2.5 Inaccuracies Due to Node Location

The reasons for node location inaccuracies and their types are discussed in Section 3.2.4. The effects of these errors will be analyzed. If the errors lead to a large number of incorrect results, the algorithms will need to have some mechanisms to overcome such errors. The effects of these errors on qualitative and quantitative characteristics will be observed.

5.2.6 Boundary Vagueness

Representation techniques for vague boundaries will be studied and applied to the distributed nature of WSNs. One solution could be an inner and outer boundary with a gradient between them. This gradient may be represented in fuzzy terms. Spatial and temporal characteristics of regions (such as their intersection) may be determined using computations based on these fuzzy terms. There must be some topological relationship between these outer and inner boundaries. For example, the inner boundary should always be *inside* the outer boundary, although they may *meet*.

5.2.7 Boundary Anomalies

Regions that are fronts in a WSN rather than being fully enclosed in the WSN are special problems. Perimeter nodes that are in the region have to be able to determine that they are boundaries for the region with respect to the WSN. Perimeter determination strategies already exist in WSN literature [22] and these can be employed, or be assumed to exist in the underlying architecture.

Regions that have one-node thick sections can be identified by nodes that have rotationally-ordered neighbors that alternate between being in the region and being outside the region. The concern will be to detect such a feature and to see how it affects the computation of the region's characteristics. The algorithms will need to be robust enough to handle these anomalies.

5.3 Work Completed

Ptolemy was the first tool to be tested. The simulation in Ptolemy involved implementing the unit disk graph and greedy triangulation (Section 5.2.1) in order to determine which nodes are the boundary nodes. The result using greedy triangulation is shown in Fig. 7.

With 10 to 20 nodes, Ptolemy's performance was good, giving results in a few seconds. With 100 nodes, the simulation takes approximately 30 minutes. In order to obtain faster simulation times, the Repast simulator will be tested next.

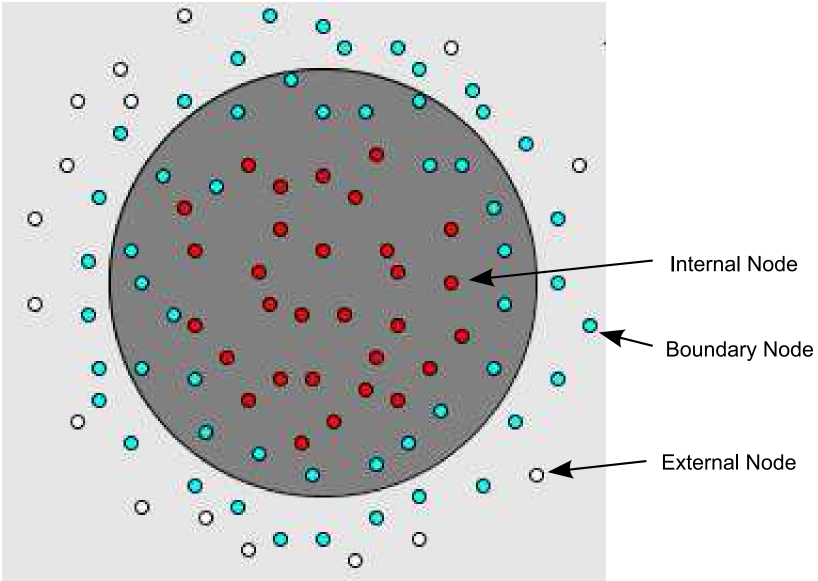

Figure 7: Boundary Nodes using Greedy Triangulation in Ptolemy

In the simulation, each node periodically updated its neighbors of its status. The algorithm for the unit disk graph was relatively simple. The node has a table of its neighbors and their readings. Whenever a new message arrives, it adds it to the table and checks whether the new message causes a change in its boundary status.

In contrast, triangulation is a computationally expensive process. Therefore, the triangulation is performed only when a message from a node with a new location is received. Then a check is made among the readings of the triangulation neighbors to determine whether the node is a boundary node.

6 Research Schedule

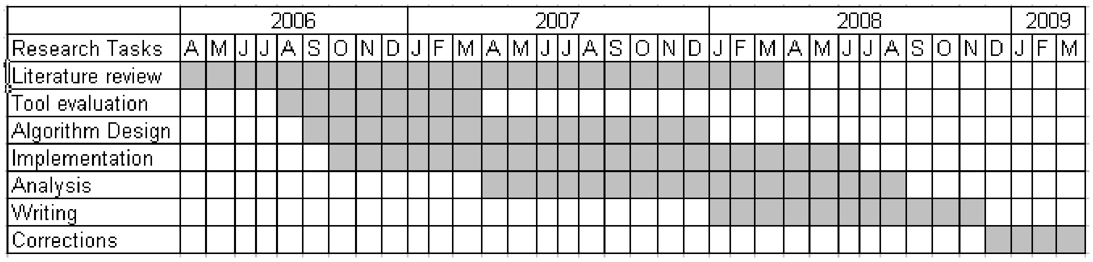

Figure 8: Work plan

Date for Submission of Final Draft: 31 December 2008

References

[1] S.W. Arms, C.P. Townsend, D.L. Churchill, J.H. Galbreath, and S.W. Mundell. Power Management for Energy Harvesting Wireless Sensors. In *SPIE International Symposium on Smart Structures & Smart Materials*, March 2005.

[2] David Braginsky and Deborah Estrin. Rumor Routing Algorthim for Sensor Networks. In *WSNA '02: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 22–31, New York, NY, USA, 2002. ACM Press.

[3] Zhao Cheng and Wendi B. Heinzelman. Flooding Strategy for Target Discovery in Wireless Networks. In *MSWIM '03: Proceedings of the 6th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile systems*, pages 33–41, New York, NY, USA, 2003. ACM Press.

[4] Krishna Chintalapudi and Ramesh Govindan. Localized Edge Detection in Sensor Fields. *Ad Hoc Networks*, 1(2-3):273–291, 2003.

[5] Matt Duckham, Silvia Nittel, and Mike Worboys. Monitoring Dynamic Spatial Fields Using Responsive Geosensor Networks. In *GIS '05: Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*, pages 51–60, New York, NY, USA, 2005. ACM Press.

[6] Adam Dunkels, Niclas Finne, Joakim Eriksson, and Thiemo Voigt. Run-Time Dynamic Linking for Reprogramming Wireless Sensor Networks. In *Proceedings of the Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, Boulder, Colorado, USA, November 2006.

[7] Max J. Egenhofer and Robert D. Franzosa. On the Equivalence of Topological Relations. *International Journal of Geographical Information Systems*, 9(2):133–152, 1995.

[8] Ben Greenstein, Eddie Kohler, David Culler, and Deborah Estrin. Distributed Techniques for Area Computation in Sensor Networks. In *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pages 533–541, Washington, DC, USA, 2004. IEEE Computer Society.

[9] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In

HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.

[10] Qin Huaifeng and Zhou Xingshe. Context Aware Sensornet. In *MPAC '05: Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad-hoc Computing*, pages 1–7, New York, NY, USA, 2005. ACM Press.

[11] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 56–67, Boston, MA USA, 2000.

[12] Guang Jin and Silvia Nittel. NED: An Efficient Noise-Tolerant Event and Event Boundary Detection Algorithm in Wireless Sensor Networks. In *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, page 153, Washington, DC, USA, 2006. IEEE Computer Society.

[13] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.

[14] Wolfgang Kellerer, Christian Bettstetter, Christian Schwingenschlögl, Peter Sties, Karl-Ernst Steinberg, and Hans-Jörg Vögel. (Auto)Mobile Communication in a Heterogeneous and Converged World. *IEEE Personal Communications Magazine*, 8(6):41–47, December 2001.

[15] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. Geographic Routing Made Practical. In *Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI '05)*, Boston, MA, USA, May 2005.

[16] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. On the Pitfalls of Geographic Face Routing. In *Proceedings of the Third ACM/SIGMOBILE International Workshop on Foundations of Mobile Computing*, pages 34–43, New York, NY, USA, September 2005. ACM Press.

[17] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In *Proceedings of the 4th ACM*

International Symposium on Mobile Ad Hoc Networking and Computing (MOBI-HOC), pages 267–278, New York, NY, USA, June 2003. ACM Press.

[18] Geoff Leach. Improving Worst-Case Optimal Delaunay Triangulation Algorithms. In *4th Canadian Conference on Computational Geometry*, 1992.

[19] Xiang-Yang Li, Gruia Calinescu, Peng-Jun Wan, and Yu Wang. Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.

[20] Nancy A. Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[21] Alan M. Mainwaring, David E. Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[22] Fernando Martincic and Loren Schwiebert. Distributed Perimeter Detection in Wireless Sensor Networks. WSU-CSC-NEWS/03-TR03, July 2004.

[23] Rex Min and Anantha Chandrakasan. MobiCom Poster: Top Five Myths about the Energy Consumption of Wireless Communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(1):65–67, 2003.

[24] Wolfgang Mulzer and Günter Rote. Minimum Weight Triangulation is NP-Hard. In Nina Amenta and Otfried Cheong, editors, *Proceedings of the 22nd ACM Symposium on Computational Geometry, June 5-7*, pages 1–10, New York, NY, USA, 2006. ACM Press.

[25] Michael J. North, Nicholson T. Collier, and Jerry R. Vos. Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16(1):1–25, January 2006.

[26] Robert Nowak and Urbashi Mitra. Boundary Estimation in Sensor Networks: Theory and Methods. In Feng Zhao and Leonidas J. Guibas, editors, *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, volume 2634 of *Lecture Notes in Computer Science*, pages 80–95. Springer, 2003.

[27] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[28] Jan M. Rabaey, M. Josie Ammer, Julio L. da Silva, Danny Patel, and Shad Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7):42–48, 2000.

[29] Mohammad Rahimi, Hardik Shah, Gaurav Sukhatme, John Heidemann, and Deborah Estrin. Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 19–24, Taipai, Taiwan, May 2003. IEEE.

[30] Jan Remy and Angelika Steger. A Quasi-Polynomial Time Approximation Scheme for Minimum Weight Triangulation. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 316–325, New York, NY, USA, 2006. ACM Press.

[31] Karim Seada, Ahmed Helmy, and Ramesh Govindan. On the Effect of Localization Errors on Geographic Face Routing in Sensor Networks. In Ian F. Akyildiz, Deborah Estrin, David E. Culler, and Mani B. Srivastava, editors, *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pages 312–313. ACM, 2003.

[32] Sameer Sundresh, WooYoung Kim, and Gul Agha. SENS: A Sensor, Environment and Network Simulator. In *The 37th Annual Simulation Symposium (ANSS37), Arlington, VA, April 21*, 2004.

[33] Gerard Tel. *Introduction to distributed algorithms*. Cambridge University Press, New York, NY, USA, 1994. 94016147 Gerard Tel. Includes bibliographical references (p. 512-524) and index. 1. Introduction: Distributed Systems – Pt. 1. Protocols. 2. The Model. 3. Communication Protocols. 4. Routing Algorithms. 5. Deadlock-free Packet Switching – Pt. 2. Fundamental Algorithms. 6. Wave and Traversal Algorithms. 7. Election Algorithms. 8. Termination Detection. 9. Anonymous Networks. 10. Snapshots. 11. Synchrony in Networks – Pt. 3. Fault Tolerance. 12. Fault Tolerance in Distributed Systems. 13. Fault Tolerance in Asynchronous Systems. 14. Fault Tolerance in Synchronous Systems. 15. Stabilization – Pt. 4. Appendices. A. Pseudocode Conventions. B. Graphs and Networks.

[34] Robert Tobias and Carole Hofmann. Evaluation of Free Java-libraries for Social-scientific Agent Based Simulation. *Journal of Artificial Societies and Social Simulation*, 7(1), 2004. available at http://ideas.repec.org/a/jas/jasssj/2003-45-2.html.

[35] Andrew Wang, SeongHwan Cho, Charles Sodini, and Anantha Chandrakasan. Energy Efficient Modulation and MAC for Asymmetric RF Microsensor Systems. In *ISLPED '01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, pages 106–111, New York, NY, USA, 2001. ACM Press.

[36] James R. Whiteley and James F. Davis. Qualitative Interpretation of Sensor Patterns. *IEEE Expert: Intelligent Systems and Their Applications*, 8(2):54–63, 1993.

[37] Michael Worboys and Matt Duckham. *GIS - A Computing Perspective*. CRC Press, 2004.

[38] Mike F. Worboys and Matt Duckham. Monitoring Qualitative Spatiotemporal Change for Geosensor Networks. *International Journal of Geographical Information Science*, 20(10):1087–1108, 2006.

[39] Wei Ye, John Heidemann, and Deborah Estrin. An Energy-Efficient MAC protocol for Wireless Sensor Networks. In *21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1567–1576, New York, NY, USA, June 2002. USC/Information Sciences Institute, IEEE.

[40] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. Technical report, University of California at Los Angeles, May 2001.

[41] Feng Zhao and Leonidas Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Elsevier/Morgan-Kaufmann, Amsterdam, 2004.