In-network Distributed Event Boundary Computation in Wireless Sensor Networks: Challenges, State of the art and Future Directions

Farhana Jabeen[1], Sarfraz Nawaz[2]
[1] COMSATS Institute of Information Technology, Islamabad, Pakistan
[Email: *farhanakhan@comsats.edu.pk*]
[2] Computer Laboratory, University of Cambridge, Cambridge, United Kingdom
[Email: sarfraz.nawaz@cl.cam.ac.uk]
*Corresponding author: Farhana Jabeen

Received June 3, 2013; revised September 11, 2013; accepted October 23, 2013; published November 29, 2013

Abstract

Wireless sensor network (WSN) is a promising technology for monitoring physical phenomena at fine-grained spatial and temporal resolution. However, the typical approach of sending each sensed measurement out of the network for detailed spatial analysis of transient physical phenomena may not be an efficient or scalable solution. This paper focuses on in-network physical phenomena detection schemes, particularly the distributed computation of the boundary of physical phenomena (i.e. event), to support energy efficient spatial analysis in wireless sensor networks. In-network processing approach reduces the amount of network traffic and thus achieves network scalability and lifetime longevity.

This study investigates the recent advances in distributed event detection based on in-network processing and includes a concise comparison of various existing schemes. These boundary detection schemes identify not only those sensor nodes that lie on the boundary of the physical phenomena but also the interior nodes. This constitutes an event geometry which is a basic building block of many spatial queries. In this paper, we introduce the challenges and opportunities for research in the field of in-network distributed event geometry boundary detection as well as illustrate the current status of research in this field. We also present new areas where the event geometry boundary detection can be of significant importance.

Keywords: Spatial Analysis; Wireless Sensor Network; Agriculture Monitoring; Distributed Algorithms; Edge Computation; Event geometry; Boundary Computation.

http://dx.doi.org/10.3837/tiis.2013.11.015

1. Introduction

Wireless sensor networking is an emerging technology for monitoring temporal and spatial behavior of transient physical phenomena, for example, a moving band of rain and a shape-shifting region of low temperature [1]. A wireless sensor network allows each sensor node to associate with each sensed measurement the location where the measurement was taken, the time at which the measurement was taken and the knowledge that what does the measured reading represent. As an example of the usefulness of this information, consider the following context. Efficient water management is a major concern for farmers of many crops. Imagine that a farmer has deployed sensor nodes and is interested in parts of a field where the soil moisture has dropped below a certain threshold so that only those parts can be irrigated [2] given the limited water supply. In addition, sensor nodes can also help in frost monitoring [3] and fighting fungal disease in the fields [4,5]. Such diseases tend to spread under certain temperature and humidity conditions. The raw data collected from the sensor nodes enables the farmer to identify regions of interest. In these deployments, nodes sense the physical features (e.g., soil moisture, temperature) and communicate with each other to send this information from the fields to the farmer. A WSN, therefore, allows the farmer to get a real-time digital picture, in the form of sensed measurements of the physical world. By combining results at different time instances, it is possible to obtain the time evolution of a physical phenomena (i.e., event region). This paper specifically considers those events that can be modeled as having a boundary, or edge. Examples include temperature gradients, variations in levels of measurable quantities such as light intensity, temperature, chemical concentration etc.

In WSN literature, there are broadly two strategies for performing analysis of the data collected by sensor nodes in a WSN. The first strategy requires each sensor node in the WSN to send each sensed measurement to a central basestation without performing any local processing whereas the second strategy involves local processing of the sensed measurements by sensor nodes as they route data packets towards the basestation. The former strategy is often referred to as warehousing (or out-of-network processing, centralized processing) and the latter as in-network processing.

1.1. Out-of-Network Processing Approach

Out-of-network event (i.e., physical phenomena) detection requires that each sensor node transmits every sensed value to a central base station. Once the data is collected at the base station, classical algorithms can be applied to the collected sensed data for estimating the event regions, after which detailed analysis becomes possible (for example to compute size, number and shape of event regions, their locations and their topological relationship with application-specific geometries). However, this approach is prohibitively expensive and does not scale well. For a typical sensor node, communication is more expensive than computation [6]. As described in [7], in a noise-free environment transmitting 1 Kb of data across a distance of 100 meters costs around 3 joules. However, a general purpose processor with 100 MIPS/W consumes around the same amount of energy to execute around 3 million instructions. Therefore, reducing communication and increasing processing inside the network increases network lifetime. For network scalability, it is also preferable that, instead of sending raw sensor data, high-level information is returned by the network. One common strategy to achieve this is to perform data reduction (e.g., by applying predicates, computing aggregates) and filtering as early as possible in the data flow path. This approach has led to the development of generalized algorithms supporting different application scenarios (e.g., TinyDB [8]).

1.2. In-network Processing Approaches

Distributed in-network processing can significantly reduce the amount of traffic in a network. Following are the two possible techniques that make use of in-network processing approach for efficiently sending information out of network for detailed spatial analysis.

1.2.1 Event Detection Technique

In-network event detection requires a node to apply an event predicate (e.g., $moisture < \theta$) to the readings obtained from transducers on the node. Nodes satisfying the event predicate constitute an event region. Complex event detection requires readings from more than one sensing device in the node. For example in agriculture monitoring application, for controlling the spread of disease, pesticides need to be applied only if temperature and humidity conditions stand in some specific relationship (soil moisture has risen above and temperature has dropped below certain thresholds). Therefore, the characterization predicate can be a more complex algebraic expression (such as $temperature < \theta$ AND $moisture > \theta'$) requiring that more than one threshold test is applied over several readings as well as Boolean connectives to compute the final result.

This approach requires that only those nodes that are part of the event, send the information towards the gateway for performing detailed spatial analysis out of network. The gateway is responsible for creating the snapshot of the phenomena based on the received information. Such in-network processing will result in the reduction of network traffic by requiring the event nodes to send the Boolean result towards the gateway. The experimental evidence in [9] demonstrates that the cost of sending event information from each node of the network is much higher than the case where only event nodes transmit this information.

More specifically, changes in the event status obtained by a group of sensor nodes can be used to characterize the life cycle of an event of interest by the way its geometry changes. At different points in time, event predicate will change in its truth value at different nodes depending on their location with respect to the event. Event detection can also result in multi-element regions, i.e., one consisting of disjoint component regions. Furthermore, the interior of an event region may contain one or more disjoint regions comprising of non-event nodes, termed as *holes*. In case of WSN, at each evaluation episode, the event region may comprise of single-element (with zero, one or more holes) or multi-element (with zero, one or more holes) geometries. **Figure 1(a)**, shows the fields $f1, f2..., f10$ (i.e., application-specific geometries) in camalie vineyard [10], and **Figure 1(b)** shows an example of a WSN over **Figure 1(a)** depicting multi-element event regions M and T along with application-specific geometries. M represent the region where $moisture > \theta'$ and T represents the region satisfying predicate $temperature < \theta$. Both elements of M and T overlap each other. One of the M region elements lying in field $f4$ contains a *hole* comprising of six nodes that are not part of event region M. It can be seen that the *hole* inside the event region results in an interior boundary. There are two kinds of boundary nodes in the M region element lying in field $f4$, shown in **Figure 1(b)**. The nodes that are lying on the boundary of the event geometry constitute *outer boundary* and the nodes that are lying on the boundary of a hole constitute *interior boundary*. Both the *interior* and *outer* boundary separates the event nodes from the non-event nodes.

Given the large size of the network, or large size or large number of disjoint event regions inside WSN, the cost and time required for sending information from event nodes might be quite high. For example, there might be a time requirement for estimating the open out of the contamination in soil to specific sensitive location. Similarly, for frost protection, some farmers make use of wind machines, while others, rely on irrigation along with wind machines to adequately modify temperature. These frost/freeze protection systems are expensive. Therefore, these should be used only if the temperature reaches a critical threshold. This stresses on the requirement to support some efficient in-network processing algorithms which further reduce the network traffic and hence reducing the response time. One of the solution proposed in the literature is to compute the boundary of event region based on the event information.

1.2.1 Event Boundary Detection Technique

The main benefits behind computing the boundary of an event region using distributed in-network processing are two-fold. First, it reduces the amount of packets to be relayed out of the network (i.e., requiring only the boundary nodes to send information towards gateway) and hence results in network longevity and scalability. Second, it allows to collect the information at higher rate from the network. Therefore, the boundary information can be used to track events entering or leaving a region thus allowing for more fine grained analysis of the phenomena. It will not only help in providing appropriate services to the affected subregion in case of emergency but will also be useful for detailed analysis of the phenomena. For example, regular spray of pesticides needs to be stopped in regions with high temperature or low moisture as it endangers the crop health.

The problem of event boundary detection can be formulated as follows: given a set of measurements from sensor nodes located at different locations and an event characterization predicate, return a description of the boundary of the event in a distributed manner. In other words, the boundary of event region separates the sensor nodes that satisfy the characterization predicate from those that do not. The boundary of an event geometry (i.e., the spatial shape of event) G_e is defined as the set of nodes lying on the boundary of G_e, denoted by $B(G_e)$. $B(G_e)$ describes the shape and the location of the event. Geometry occupies a position in space defined by its interior, boundary and exterior. A boundary detection algorithm allows a node to determine if it lies on the boundary or the interior of event geometry. This allows the WSN to be treated as a distributed spatial database containing information for both event and application-specific geometries as shown in **Figure 1b**. Similar to the aforementioned in-network processing approach, this approach also requires the base station to compute the detailed snapshot of the WSN based on the boundary information (i.e., all possible *outer* boundary and *inner* boundaries) of the event geometry to perform detailed spatial analysis.

In this state of affairs, if environmental scientists were to try and seek guidance in the literature as to which, among the proposed event region boundary detection approaches, would perform better for a planned deployment, the absence of such literature will act as a major drawback in taking the right decision. While existing survey studies [11] algorithms for detecting holes boundary [12] (which exist in WSN because of nodes failure or arbitrary deployment), and estimating the network boundary [13]. Our work targets in-network processing techniques for detecting boundaries of event regions. These techniques are classified and discussed in detail providing a complete overview of challenges and future directions. There also exist studies that deal with coverage problem in WSN [14]. These algorithms [11,12,13,14] focus only on the physical network topology. In this paper we are concerned with the detection of event region and its boundary. Event regions are representations of transient phenomena determined by physical properties (e.g., humidity, or temperature) that can be sensed by the nodes. These event regions are assumed both not to pre-exist the WSN deployment and to change independently, possibly often, for the duration of that deployment.

The remainder of the paper is organized as follows: Section 2 presents the challenges associated with the distributed in-network boundary detection of the event region; Section 3 surveys the work related to event region boundary detection; and Section 4 highlight the current research trends related to boundary detection in WSN. Section 5, concludes the paper.

2. Challenges

Event boundary detection using in-network processing techniques in sensor networks raises research issues at several levels. Some of the challenges are as follows:

2.1 Distributed Data

In a WSN, each sensor node has limited coverage and connectivity. Because of the limited coverage, each node is only aware of a small part of the event geometry that lies within its sensing range. Complete information regarding the event region is distributed throughout the network. Classical boundary detection techniques cannot be used directly as they are usually applied to centrally stored data. Therefore, for in-network computation of event geometry and its boundary, sensor nodes require information from neighboring nodes. In a WSN, energy is valuable because it is scarce. Sensor nodes only have finite energy reserves drawn from batteries that cannot be easily replaced. In certain applications where nodes are deployed in inaccessible locations, it may not be possible to replace batteries [15, 16]. This makes the event region boundary detection problem challenging because, for in-network processing, any practical solution must be a localized algorithm (i.e., requiring information from 1-hop neighbors only). As the size and shape of the transient phenomena varies over time and space, each node is required to periodically reconsider its event-membership status from sensed data at the relevant evaluation instances.

2.2 Noise

Noise is usually unwanted faulty measurements reported by sensor nodes due to the following reasons: (i) hardware/software fault, (ii) some sensor nodes might be exposed to extreme environment and report extreme values (reasons include arbitrary deployment etc.). Noise in the environment may result in limiting the achievable accuracy of estimated event boundary. The increase in fault probability decreases the achievable accuracy in WSN. Because of noise, a non-event node may declare itself as an event node. Therefore, while designing a boundary detection scheme for event regions, care must be taken to detect and suppress faulty measurements.

2.3 Deployment

In WSN-based environmental monitoring applications, it is challenging to design a precise deployment. The reason is accurate event detection depends on a number of factors including the type and quality of the sensor nodes and the nature of the terrain. The type of event to be detected also varies from application to application. For dynamic, transient phenomena, the size and shape of the event region varies over time and space. Thus fine-grained event boundary estimation can only be achieved if a large number of nodes are deployed. Otherwise, one has to compromise on the precise shape and size of the event region.

Although the relatively low cost of the sensor nodes allows for the deployment of these nodes in large numbers, in many applications the cost of the nodes is not the critical factor compared to the deployment costs and the need to collect the right data at the right spatial granularity. Therefore, different applications require different geographic coverage with varying or uniform nodes density over the field under study.

The deployment strategies are broadly divided into regular grid and random deployments. In regular grid deployments, the sensors are placed deterministically with some regular geometric topology along grid points. This strategy is mostly used in application scenarios where the user has control over node placement, i.e., where access to the deployment site is not a concern and it is safe for humans to place sensors manually. Because of the fact, that each node has a communication radius r^c and a sensing radius r^s, in most applications nodes are usually deployed at a distance r^s apart along the regular grid points ensuring coverage of the area as well as node connectivity. In the case of a regular grid deployment, the granularity of the grid is, therefore, a function of the sensing range, the radio range and whatever fine-grained event-detection resolution that can be afforded. Random deployments allow for the locations of sensor nodes to be not planned a priori: the sensors may be air-dropped, scattered using a vehicle or any comparable means [17].

2.4 Aggregation of Information

After the evaluation of in-network boundary detection algorithm, the information about the event region boundary lies inside network in distributed fashion. As already mentioned in Section 1, the event region may comprise of a single-element region or a multi-element region with or without holes. Transmission of binary result (representing whether node is part of boundary) offers energy efficiency, but binary decisions are unreliable in noisy environments. The challenge here is how to fuse (integrate) the data efficiently, so that the base station can correctly compute the current snapshot of WSN on receiving information. This implies that efficient aggregation and routing strategy must be designed for routing the required information towards the gateway.

3. Boundary Detection Schemes

In case of out-of-network event region boundary detection, once the data is collected at the base station any classical image/object detection algorithm designed for centralized image/object boundary detection can be applied with little or no modification [18,19,20,21,22]. A WSN can be considered as an image. Just as an image is comprised of grid of pixels, a WSN is a collection of sensor nodes.

Realistic in-network boundary detection is difficult in WSN and an in-network boundary detection algorithm may lead to limited precision in the estimated boundary. Limited accuracy may occur because of the following reasons: (i) not detecting sensor nodes that are part of *true event boundary* (i.e., real/exact boundary of the event geometry) as boundary nodes, (ii) detecting nodes that lie close to the true boundary but not on the true boundary as boundary sensors or (iii) detecting nodes that are not even close to the true boundary as boundary nodes. The last two reasons contribute to the thickness of the event geometry boundary, resulting in deterioration of the accuracy. Many proposed solutions exist for in-network boundary detection, including those based on image processing, topological and geometrical techniques as well as statistical schemes.

To the best of our knowledge, no in-network boundary detection scheme presented in the literature presents the solution to the aggregation challenge. The schemes only detect the event boundary. The information regarding the event boundary lays in the distributed manner in side the network. In this paper, each of these techniques is compared according to the following parameters:

1. **Communication Cost:** This parameter studies the communication cost involved in the in-network detection of the boundary of the event region.
2. **Adjustment to Network Characteristics:** This parameter studies whether the scheme requires selection/computation of certain values (required for its processing) based on the network characteristics (i.e., network size, deployment strategy or on the percentage of expected noise inside WSN) before evaluation. For example, in most of the schemes presented in this section, a threshold value is computed and distributed among nodes inside WSN. This threshold value is used by WSN node to compute whether it is part of interior or boundary. In some schemes this threshold value varies based on the probability of noise, and network density etc. The selection of correct threshold value is critical for the accuracy of the scheme. Otherwise the scheme might end up with thick or non-smooth curve edges.
3. **Adjustment to Sensing Error:** The adjustment to noise parameter apprise whether the existing scheme incorporates any mechanism to handle the noise issue or the scheme only works with the assumption of no noise in the network.
4. **Trade-off Accuracy vs Cost:** This parameter studies the tradeoff between accuracy and cost. The parameter highlights the following: (i) experimental evidence for the network with random and uniform-grid deployment; (ii) event region shapes considered for experiments (i.e., realistic or circular/elliptical shape only); (iii) how much improvement in accuracy can be achieved by collecting data from more than 1-hop neighborhood**;** (iv) impact on accuracy by changing the network characteristics such as noise, network size, network deployment strategy etc.

3.1. Statistical and Geometrical Techniques

WSNs allow for the detection, tracking and monitoring of the physical phenomena in space. Each sensor node has a location in physical space, thereby enabling us to associate spatial properties with data. Geometric methods for boundary detection use geographical location information.

Table 1, compares the schemes based on statistical and geometrical techniques. Let set of nodes S is deployed on a two-dimensional Euclidean plane G . At each evaluation episode, sensor field S is given as $S = S_{NEN} \bigcup S_{EN}$ where S_{EN} represents event nodes and S_{NEN} represents non-event nodes. The 1-hop neighborhood $N^1(s_i)$ of a node $s_i \in S$ is the set of all sensors that are in its communication range.

A statistical approach [23] requires an event node to collect event information from its neighbors, derive a set of statistics from that information and use a Boolean decision function in order to decide whether it lies in the event boundary based on an acceptance threshold. This scheme works well for the scenario where all the sensor nodes are well calibrated, error free and uniformly deployed.

In the classifier-based approach [23], a node attempts to partition the information collected from its neighborhood into regions of distinct behavior using classification techniques. Under such scheme, each of the sensor nodes, in order to check whether it is a boundary sensor or not, collects information from the neighboring sensors and applies a linear classifier technique. An optimum line is selected which is then compared against a threshold to determine whether it is passing close to the boundary or not to declare the node as a boundary sensor. The classifier-based approach involves fitting lines at fine sample granularity resulting in a high computational overhead.

Jitender et al. [24] proposed the *Interior Point* (IP) algorithm to discover boundaries in uniformly and randomly distributed WSNs. The authors assumed that each node must have at least three neighbors in its radio range. This approach consists of two algorithms: IP and *ChooseGoodneighbors* (CGN). In order to detect boundary nodes, IP requires nodes to broadcast their location information to their neighbors. The IP algorithm confirms whether a node is in the radio range of three neighbors. The accuracy of the algorithm depends on the selection of three best neighbors. For this purpose, the CGN algorithm is responsible for intelligently selecting four neighbors that are pairwise neighbors of each other. CGN selects neighbors that are close to a node and possibly surround it. Experimental evidence is derived using size of neighborhood and the network as metrics in both random and uniform grid deployments.

Tangent fit (T-fit) [25] is another localized boundary detection technique based on geometric rules and trigonometry. In order to detect boundary nodes, it requires event nodes to broadcast their location information to neighboring nodes. On receiving messages from event nodes, a node s_i makes itself the origin of a circle centered at itself and partitions its neighboring event nodes into four quadrants. The boundary-detection statistic is then formulated in terms of the number of quadrants in which neighboring event nodes are found.

If no neighboring event nodes are found in any quadrant or if they are found in all four quadrants, then s_i declares itself *not* to be a boundary node. If neighboring event nodes are found in one quadrant only, s_i declares itself to be a boundary node. If neighboring event nodes are found in two or three quadrants, s_i computes the angle formed by itself at the origin and its two farthest neighbors in the two populated quadrants (or in the diagonal quadrants, in the case of three populated quadrants). If computed angle is less than $180°$, s_i declares itself to be a boundary node. Experimental results show that the scheme performs relatively well when compared to the PR-classifier algorithm [23] both in terms of accuracy and energy efficiency. The experimental evidence suggests that the performance of the scheme increases by increasing node density even in the case of arbitrarily placed nodes. The T-fit scheme does not incorporate any error suppression scheme. Therefore, in experiments with faulty measurements at random positions it gives poor results. There is some ambiguity regarding the definition of quadrants. It is not clear how to apply the rules to the nodes that lie on the axes. Furthermore, the rule to declare itself non-event node on finding event nodes in all four quadrants is also too strict. This can result in a non-smooth boundary. For example, in case of event region with jagged boundary, it is possible that s_i finds that in any one or more quadrants there is either only one or few event nodes (which might be due to noise) and many non-event nodes. The scheme, therefore, should consider the ratio of event nodes to non-event nodes when detecting boundary nodes.

Noise-tolerated Event and Event Boundary Detection (NED) [26] supports noise suppression in event boundary detection in WSNs. NED assumes that sensing errors are independent over the WSN and noise is a white normal random variable (i.e., sensor error $\in N(0,\sigma^2)$) with fixed variance assuming that all nodes are sourced from the same manufacturer batch. In addition, it assumes that the event phenomenon is continuous. NED uses a statistical approach in which the probability density of a normal random variable concentrates around the mean value. So, for a random variable, $N(mean,\sigma^2)$, 95 % probability falls within the range $(mean+1.96\sigma, mean-1.96\sigma)$. Since the sensing error is considered normal white noise, nodes are classified as *significant*, *non*-significant *event*, *non-event* based on the threshold value and variance σ^2 of the sensing error. This classification allows the transmission of a message using variable length coding mechanism for communication efficiency. The NED algorithm makes use of a *moving mean* method for suppressing sensor faults. As the authors focused on continuous phenomena to represent sensed data in WSNs, they conducted experiments using a smooth gray-scale image. Performance analysis of the NED algorithm has not been provided in terms of varying network density and neighborhood. The experimental results show that with a large density and moderate noise, NED performs well for detecting the boundary of continuous phenomenon.

The localized fault-tolerant event boundary detection scheme [27] assumes that the set of sensor nodes with faulty measurements may contain information related to detecting events. The algorithm for faulty sensor detection is based on the *moving median method*, which requires every node to broadcast location information along with the sensed measurement to its neighbors. Median is a useful statistics, which works directly with continuous numbers rather than binary readings. Each node adds its own measurement to the ones received from neighboring nodes and computes the median of these measurements. Each node then finds the difference d between its own measurement and the calculated median. It then broadcasts d in another message to neighboring nodes. Several statistical tests are then applied to collect outliers and then to compute the resultant value, which is then compared against the threshold to determine whether the node is a boundary node.

Zhang et al. propose two algorithms based on computational geometric techniques, called *Localized Voronoi Polygon* (LVP) and *neighboring Embracing Polygons* (NEP) [28]. In LVP-based algorithm, each node computes the tentative localized voronoi using the nearest neighbor distance and direction information. Based on this information, neighbors are divided into quadrants. If neighbors are found in all four quadrants, a node declares itself a non-boundary node. If a node cannot find neighbors in any quadrant, then it checks for neighbors in the *assistant area* (constructed by calculating two sectors of 45 degrees each, adjacent to the specific quadrant). If the neighbor in the *assistant area* is not the nearest neighbor in that quadrant, then the node declares itself a boundary node. The LVP-based algorithm is reported to provide continuous closed curves as boundaries. In case of the NEP-based algorithm, a node sorts its neighbors according to their angle with itself (to create a *convex hull* of its neighboring nodes). After sorting, if it finds a gap less than or equal to 180° among these angles, it declares itself a boundary node. The authors have reported that, compared to the LVP-based algorithm, the NEP-based algorithm provides less accuracy.

The fault-tolerant event boundary detection scheme (FEBD) [29] is based on Bayesian theory [30, 31]. It is another distributed localized boundary detection scheme designed for WSNs. FEBD requires every node to send the outcome of its event detection predicate to all its one-hop neighbors irrespective of whether this outcome was *true* or *false*. On receiving this information from neighbours, each node counts *true* and *false* outcomes separately. A node resets its event outcome based on the majority rule (i.e., it will declare itself as event node only if count of *true* outcomes is greater than *false* outcomes). If a node calls itself an event node, then it computes a statistic to determine whether it is boundary node by comparing that to a threshold value. This threshold is defined by the user and thus should take into account any relevant deployment properties (such as the tolerance radius for boundary thickness etc.).

The majority rule scheme [31] is used by FEBD to suppress sensor errors. The performance of the scheme decreases with an increase in the number of faulty measurements, especially in low-density networks. Compared to the schemes in [23, 27], the performance of FEBD in terms of correctly detecting boundary nodes is not too much dependent on the density of the network and, gives reasonable results at low communication cost, even with faulty measurement percentage of up to 25%. Majority voting algorithm has been found to remove the random faults but may also lead to inaccurate detection of the event region boundary. It has been identified that in high-density networks, the scheme in [29] produces poor outcomes for event regions with shapes other than elliptical or circular, i.e. regions with narrow bands or acute jagged edges. **Figure 2(a)** shows example event geometry with irregular/jagged boundary and **Figure 2(b)** demonstrates the shape of sensed event geometry. Consider the **Figure 2(b)** where event nodes *1, 2, 3, 4*, have more non-event nodes compared to event-nodes in their 1-hop neighborhood. Application of *majority rule* will result in flipping their event status from event node to non-event node contributing towards the in-accuracy of the scheme.

The boundary detection method based on the statistical scheme in [23], suffers from the fact that an optimal threshold value is required for accurate boundary detection. The authors in [32] proposed a technique to determine an optimal threshold based on the *Neyman-Pearson* (NP) criterion, termed as NP fusion method. Like [23, 29], the proposed scheme requires tolerance region r value. A node is declared as boundary node if it lies at a distance less than r from the true boundary line. For decision fusion under NP criterion, each node needs event information from its neighbors. The proposed scheme works in two stages. In the first stage, a node computes its event status by applying statistics to the information including sensed measurement, least error rate and mean of k data samples. In the second stage, a node computes whether it is part of the boundary by applying statistics to the information including neighbors event information, tolerable false alarm rate calculated based on NP, tolerance region and parameters computed based on NP criterion. It is reported that the scheme works well in scenarios with high rate of location error or with high signal-to-noise ratio compared to classifier based scheme [23].

Table 1. Boundary detection schemes based on statistical and geometrical techniques

	Communication Cost	Adjustment to Network Characteristics	Adjustment to Sensing Error	Trade-off Accuracy vs Cost
Statistical scheme [23]	Each $s_i \in S$, transmits one bit event information to $(N_1(s_i))$.	Selection of correct threshold value is critical. The variations in value result from varying network density, or ratio of neighborhood to boundary width or ratio of neighborhood to faulty measurements.	No mechanism for suppression of noise	Works well with well-calibrated and uniformly deployed nodes. Performance degrades with increase in network size and decrease in neighborhood. Collecting information up to two-hop increase performance. Experiments are based on uniformly placed nodes for event region that is elliptical in shape.

Classifier -based scheme [23]	Each $s_i \in S$, transmits location & one bit event information to $(N_1(s_i))$.	The threshold value is constant.	No mechanism	Scheme works well with nodes that are well-calibrated and uniformly deployed. The performance can be improved with improving network density & considering information from two-hop neighbors. Reported that it works well as compared to other schemes in [23]. The performance is tested only over elliptical and linear boundaries.
IP[24]	Each $s_i \in S$, transmits location information to $(N_1(s_i))$	The assumptions of the scheme includes: (i) dense networks, (ii) Each node to have at-least three neighbors.	No mechanism.	Accuracy of the algorithm decreases with the increase in network size. Scheme perform well for grid deployment compared to random deployment. The accuracy increases with neighborhood more than 4 up to certain threshold.
T-Fit [25]	Each $s_i \in S_{EN}$, transmits location & one bit event information to $(N_1(s_i))$.	The threshold value is constant.	No mechanism. But the scheme allows to detect & suppress noise in terms of event nodes detected as *isolated* nodes.	This performance increases with increase in network density. It is reported to work better in comparison to Classifier scheme [23] for large density network. The experiments are based on arbitrary placed nodes for event regions circular or concave in shape.
NED[26]	Uses variable length coding mechanism. Each $s_i \in S$, transmits to $(N_1(s_i))$ either 02 bits or 32 bits of information.	The assumption is noise variance (σ^2) is fixed. The distribution of values are normal random where 95% lies in (mean -1.96 σ, mean +1.96 σ) range.	NED make use of *moving mean method* for suppressing random sensor faults.	Scheme works well in the scenarios where sensed measurements are closely related (i.e., proximity close) and represented as continuous values (instead of digit representation). Otherwise, will result in thick and non-exact edges. Experimental results are represented in pictorial form. No clear comparison with true boundary.
localized fault-tolerants event boundary detection [27].	Each $s_i \in S$, transmits location & sensed measurement to $(N_1(s_i))$.	The algorithm is sensitive to the settings of the threshold, which is based on the fault probability in WSN.	It uses *moving median method* to identify faulty nodes.	Communication cost of the *moving median method* is approx. 32 times higher than *majority voting algorithm* [29] as it requires measurement to be broadcasted. The performance is tested only over elliptical and linear boundaries. It gives reasonable results with a faulty measurement percentage up to 20% and node densities >= 30. Performance degrades with raise in faulty measurements percentage or node density.
LVP[28]	Each $s_i \in S$, transmits location information to	Works with the assumption that boundary nodes	No mechanism	The experiments are based on uniformly placed nodes. Accuracy increases by raising

	$(N_1(s_i))$. In addition, communication is required to compute localized voronoi polygon.	can be locally detected, if and only if communication range of a node is twice the sensing range.		the node density up to certain limit after which it remains constant. The experimental section does not discuss the accuracy obtained compared to true boundary.
FEBD [29]	Each $s_i \in S$, transmits event information to $(N_1(s_i))$.	The threshold value is constant and can be adjusted for true representation of boundary thickness.	FEBD make use of *majority rule* for suppressing sensor nodes faulty measurements.	The experiments are based on uniformly placed nodes with and without fault probability of up to 25% nodes for event regions circular or rectangular in shape. Reported can perform better then classifier-based scheme [23].
NP[32]	Each $s_i \in S$, transmits event information to $(N_1(s_i))$.	Works with dense networks	Provides method for noise detection and suppression. Node sets its event status after applying statistics on information received from neighbors.	The experimental section does not discuss the accuracy of scheme compared to crisp true boundary. Experiments are run over square/circular shape event region with average density of 20 nodes. Experimental results are represented in pictorial form. No clear comparison with true boundary.

3.2 Image Processing Techniques

Like a pixel value in an image, a sensor node, based on its sensed measurement, can estimate whether it belongs to the event region. **Table 2**, compares the boundary-detection schemes based on image-processing techniques.

To the best of our knowledge, the first characterization of event boundary was reported in [23] where the authors propose three different approaches: *statistical*, *image-processing* and *classifier-based*. None of these approaches has any mechanism for detection and suppression of noise. The image-processing approach treats each sensor as a pixel, thereby opening the way for the direct application of boundary detection techniques used for images, e.g., those based on computing a filtered image using convolution, *Prewitt filters* [33] etc. The image processing scheme [23] does not consider either the topology or the locations of the sensor nodes. A node maintains a separate count for the number of neighbors that satisfy and those that do not satisfy event predicates in each quadrant with the node itself as the origin. Note that if a WSN deployment is such that the required pixel-like regularity is not possible, a weighting scheme based on the number of event and non-event neighbours is presented.

Prewitt filter, is applied by every sensor node in a network to determine whether it is a boundary sensor or not [33]. The scheme requires each node to broadcast its measurement along with location information to its 1-hop neighbors. Application of Prewitt filter requires that the space around the node be divided in to eight sectors of 45 degrees each with at-least one neighbor per sector. In random deployments of sensor nodes, there is a possibility that the sensor node might not have eight neighbors. To handle such scenarios, the solution presented in the paper is that the node uses its own measurement for those sectors where no neighbors are present. When there is more than one neighbour per sector, the mean of their measurements is used for that sector. A node computes its event region participation based on the computed average value. This approach termed as *mean filter* method helps suppressing random faulty measurements.

3.3. Topological Techniques

Techniques in this category make use of connectivity information and do not require knowledge about node locations. **Table 3** compares the schemes based on topological techniques.

Table 2. Boundary detection schemes based on image processing techniques

	Communication Cost	Adjustment to Network Characteristics	Adjustment to Sensing Error	Trade-off Accuracy vs Cost
Prewit Filter [23]	Each $s_i \in S$, transmits location & one bit event information to $(N_1(s_i))$.	Selection of correct threshold value is critical. The variations in value result from varying network density, or ratio of neighborhood to boundary width or ratio of neighborhood to faulty measurements.	No mechanism	Performance degrades with increase in network size, decrease in neighborhood size & increase in noise. Works well with uniformly-deployed nodes. Reported that it works well as compared to statistical scheme [23]. The performance is tested over elliptical and linear shape boundaries.
Prewit Filter [33]	Each $s_i \in S$, transmits location & sensed measurement to $(N_1(s_i))$.	Selection of correct threshold value is critical. The variations in value result from *varying* network parameters such as density etc.	*Mean Filter Method*	Mean square error increases with increase in communication range. This scheme performs well in scenairios with high node density, low communication range and event region having elliptical shape boundaries.

Wang et al. [34] proposed a distributed algorithm for boundary detection of WSN. The algorithm assumes that there are holes in the network. Reasons for the existence of holes include arbitrary deployment and dead and faulty nodes. In its current form, the algorithm does not detect event or its boundary but can be modified to support detection of event geometry with holes. The algorithm works by creating a shortest path tree in the network and finding the *cut* nodes. The shortest path terminates because of the holes in the network. *Cut* nodes are defined as the set of nodes where the shortest paths of distinct homotopy type meet after passing around holes. By using the *cut* nodes, *shortest cycle* enclosing the composite hole is computed. To find the outer boundary (i.e., boundary of WSN), a flood is generated from the *shortest cycle* to compute the extreme nodes (part of the boundary of WSN). Later on some more statistics are applied and communication is carried out to refine the inner and outer boundaries.

Nowak and Mitra [35] proposed a hierarchical processing strategy using a cluster-based scheme for boundary detection in WSNs. The whole sensor field is first divided into four quadrants and each quadrant is then recursively divided into 4 sub-quadrants of equal size until some maximum resolution is reached. Then cluster formation occurs. Each of the sensor nodes in the sub-quadrant is then responsible for transmitting their original measurements to their sub-quadrant cluster head. The cluster head is then responsible for computing the average and other statistical methods before transmitting its estimates regarding the sub region to the cluster head up in the hierarchy. A quad tree is used for representing this hierarchical structure. The cluster head on the higher level then performs some further processing in order to determine the sub-partition that provides an approximation to the boundary. The algorithm does not involve any mechanism for detection of noise. However, since the average of the measurements is used instead of the original measurements, there is suppression of noise to some extent. This scheme considers the topology as well as the locations of sensor nodes. Under this scheme, upper bounds are set on the *Mean Square Error* (MSE) of the estimator based on the smoothness of the curve. The authors concentrate on the trade-off between the MSE and the communication cost as a function of node density. The MSE increases with an increase in node density.

Liao et. al. proposed a *Non-uniform distribution Self-Organization Overlapping Algorithm for Clustering* (Nu-SOAC) for boundary detection of WSN and event geometry [36]. Cluster formation occurs based on energy and connection density information. Event nodes compute a weight based on energy and connection density information received from event neighbors. Nodes with highest weight in *k-hop* neighborhood are elected as cluster heads. These cluster heads are responsible for cluster formation by sending the join request to *k-hop* neighbors. Nodes accept the request from the cluster head with higher weight. After the formation of clusters, the boundary nodes of the clusters are identified. The nodes that are part of the overlapping clusters then help in fusing clusters boundaries to compute the boundary of the event region. This work does not consider the scenarios where there are one or more holes or multi-element event geometries.

Jaffer et al. [37] used an autonomous agent based approach for event boundary detection in WSNs. The main objective of this work is to reduce the communication cost by improving transmission efficiency. Initially, event nodes generate agents based on a preset threshold value. The node with the agent requests a response from non-event neighboring nodes. The agent makes a decision about the selection of the next boundary node on receiving the response from its neighbors. On finding the first boundary node, the agent generates a child agent. Both the agent and the child agent then start moving around the boundary in the opposite directions until they meet. An agent stores the boundary nodes that it visits in its boundary stack. The threshold value used for generation of an agent is not constant and is dependent on the phenomenon size. This scheme only considers the topology of the sensor nodes deployed in the field. According to the experimental results, the scheme performs well for networks with high node density. Furthermore, the efficiency of the algorithm increases as an inverse function of the event region radius. No discussion is provided regarding the stack size of the agent used to keep boundary-related information, on the cost associated with agent movement from one node to another and additional communication cost overhead due to multiple agent's production.

Table 3. Boundary detection schemes based on topological techniques

	Communication Cost	**Adjustment to Network Characteristics**	**Adjustment to Sensing Error**	**Trade-off Accuracy vs Cost**
Topology-based scheme [34]	Communication is required to compute short path tree in WSN, and to compute the short distance from the boundary of the hole towards the boundary of WSN. Additional cost is involved in terms of refining holes boundary and WSN	Works with the following assumptions: (i) There exists a hole in WSN, and (ii) Minimum size of the hole is known.	No mechanism	The scheme presented work only with scenarios where network contains one or more holes. The scheme requires average node degree of at least 7. The performance is tested with scenario's where WSN is rectangular in shape and holes inside WSN elliptical/circular shape. Accuracy compared to true boundary is not discussed.

	boundary information.			
Cluster-based scheme [35]	Each $s_i \in S$, transmits sensed measurement to $(N_1(s_i))$. Construction and maintenance of Quad-tree and clusters. Information transmission by nodes towards cluster heads and by cluster heads towards their heads up in the hierarchy.	Upper bounds are set on the MSE of the estimator based on the smoothness of the curve. For recursive distribution of WSN field maximum limit is required.	No specific mechanism. The cluster heads computes average which results in suppression of error to some extent.	The scheme provides better accuracy for low and medium density networks but at the cost of complex regularization of the hierarchical tree-based estim-ation method. The assumption is nodes are uniformly deployed. It is based on a hierarchical polling which implies high cost in terms of cluster formation, head select-ion, its maintenance, as well as long transmission distances.
Nu-SOAC [36]	Each $s_i \in S$, transmits energy, connection density information to $(N_1(s_i))$. Additional commun-ication cost is involved in cluster formation and clusters boundary fusion	Works with the following assumptions: (i) Channels and sensor are faultless, and (ii) All nodes are having similar communication range.	No mechanism	High overhead due to cluster formation and maintenance. For higher accuracy of the scheme, information is required from 2-hop neighborhood. The experimental section does not discuss the accuracy compared to true boundary. Performance is tested with event region rectangular in shape.
Agent-based scheme [37]	The node on which agent resides polls neighboring nodes requesting response from event nodes. In addition, communic-ation cost is involved related to movement of agent from one node to other.	Selection of correct threshold value is critical. Selection of which depends on size of the event region. The value is used to control the number of agents to be generated.	No mechanism	The scheme might result in high overhead due to generation of multi-agents. Performance is found highly suspectable both in terms of accuracy and false positives in case of network with low density. Efficiency of the algorithm increases with the increase in circumference of event-region.

Image processing is usually dependent on information from large neighborhood for computing pixel value. In WSN applications, this might not be possible because of the following reasons: (i) limited neighborhood (reasons include low density network, arbitrary/random deployment, or lossy communication links, hardware problems etc.,), and (ii) information gathering from large neighborhood is expensive. These schemes perform well in scenarios with location information, high node density, low communication range and uniformly deployed networks with less or no noise.

Schemes using topological techniques make use of connectivity information and do not require knowledge about node locations. Most of these schemes require the nodes to be organized into network wide structures like trees or clusters involving additional costs. Schemes using topological techniques work well with low and medium size networks having no or low noise. The schemes based on geometrical techniques require location information and works efficiently with medium to large size networks having no or moderate noise as compared to schemes based on image processing and topological techniques.

4. Future Research Scopes and Open Issues

The spectrum of applications for WSNs spans multiple domains. In environmental sciences, in particular, they are becoming an essential technology for monitoring natural environment and for modeling the dynamic behavior of transient physical phenomena over space. The distributed detection of event region is not only useful for agriculture monitoring but can also be applied to other environmental scenarios as well. Examples include, removing a pollutant plume, fire fighting or performing a rescue action after an earthquake.

In existing literature, the proposed boundary detection schemes have been tested with event regions with Platonic (such as circles, squares and ellipses) shapes instead of considering a realistic shape (i.e., with irregular/jagged) that occurs in nature (e.g., consists of events reported in [38, 39]). In addition, most of the techniques have not been tested using real sensor hardware or simulators/emulators available for WSN. This acts as a major hindrance in taking an informative decision.

There is a dearth of literature related to efficiently transmitting boundary information of event region out of network. The event region may comprise of either a single-element or multiple-elements. In addition, such multiple-element geometries may also encompass regions that are not part of the region, for example, region with a hole in the middle. Monitoring and tracking multiple event geometries over time is a much more complicated task. To handle such situations, an energy-efficient aggregation strategy is required (to efficiently transmit boundary information to the gateway) that not only accounts for the transient nature of the physical phenomena but also for the fact that such event geometries might move in intersecting/overlapping patterns. In addition to this, some applications require the results to be produced in near real time.

In recent research, *Sensor Network Query Processors* (SNQP's) have been demonstrated to be an effective and efficient means of interacting with a sensor network in data collection tasks [8, 40]. SNQP's allow for energy-efficient in-network evaluation of declarative data analysis queries. Thus spatial analysis over sensor networks can be built on established in-network distributed query processing techniques. However, the emphasis should be to concentrate more on the spatial aspects of the data that are not adequately addressed in existing SNQPs. Such spatial SNQP's will allow the farmer to pose declarative spatial queries and get fine-grained results out of the network.

For performing in-network spatial analysis, the spatial queries that are of interest to a farmer can be divided in to three categories: (1) finding topological relationships; (2) deriving new geometries; and (3) computing the spatial features of the geometries including $Area$, $Perimeter$ etc. Topological relationships (like $AreaInside$, $Disjoint$, $Intersects$) yield a Boolean decision to signify if a certain relationship holds between two spatial objects in space or not. In our agriculture monitoring example scenario, the farmer might be interested in knowing the existence of a topological relationship (whether M $Intersects$ $f5$ in a vineyard in **Figure 1(b)**) to take certain decisions. For example, to remove leaves and expose grapes to more sunshine, or to change the schedule for using pesticides.

Apart from reducing network traffic and increasing network longevity, in-network evaluation of such topological relationship queries allows sending only fine-grained results (i.e., Boolean result) out of network to the user. In-network computation of topological relationships requires the availability of event and application-specific geometries inside the network (including information about its boundary, interior and exterior) along with the definition of distributed spatial algebra and implementation of topological operators algorithms. Since the nodes in a WSN and the communication edges formed by them map naturally to a finite set of points and a finite set of line segments respectively, an algebraic approach to spatial analysis is intuitively possible. In-network evaluation of spatial-valued operators (such as $Plus$, $Minus$, $Intersection$) also requires information about geometries to derive new geometries (e.g., $M \quad Intersection \quad f5$). This highlights the importance of in-network distributed boundary detection algorithms. These algorithms provide the informati/[

‘[on needed for the higher-level tasks such as derivation of geometries and topological relationships. From the aforementioned examples, it can be seen that the detection of event region boundary is the first step towards distributed in-network spatial analysis.

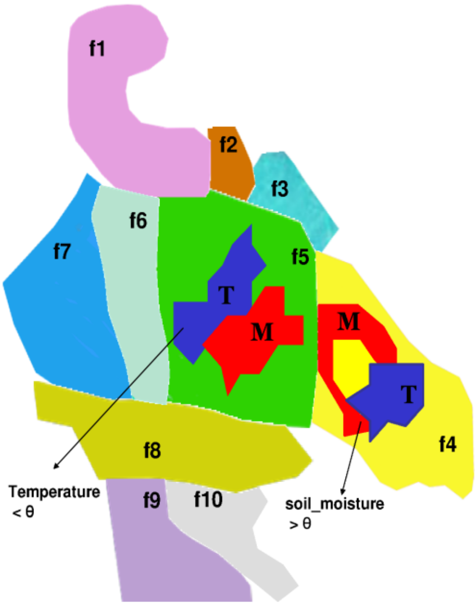
For computing topological relationships, clear and crisp boundary detection of the event regions is required [40, 41, 42]. Achieving ideal boundary is difficult in sensor networks. Most of the existing techniques do not put emphasis on the efficiency of the proposed technique in terms of boundary thickness. Boundary thickness occurs due to detection of fraction of sensor nodes as boundary sensors that are not part of the true boundary. As mentioned earlier in Section 3, two types of errors contribute to boundary thickness. Only few schemes (like [29]) have proposed approaches for controlling the thickness of the boundary during the detection process.

In [44], the authors propose logical neighborhoods for sensor nodes. For example, a logical neighborhood of a node is defined as nodes where temperature is higher than a certain threshold and that are at a maximum of 5-hops away. In such scenarios, a scheme can be designed where the boundary nodes of each element are responsible for keeping distance information to other elements. Such a scheme will be efficient because it will limit the amount of broadcast by suppressing the logical neighborhood creation requests from interior nodes of the region-element. The reason is under this scheme, boundary nodes keep distance information to other elements. True boundary detection will allow the nodes in the WSN to maintain and access information about logical neighborhood at a much lower cost. It will also allow efficient computation of other features including *distance* between two or more elements and *perimeter* of the region.

This discussion shows that there is a need for more sophisticated in-network event and boundary detection techniques. Energy efficiency and crisp edges are key requirements for such techniques to be useful for distributed spatial analysis in wireless sensor networks.

5. Conclusions

This paper highlighted some of the major challenges that are associated with the precise estimation of boundaries of event regions over wireless sensor networks. It also presented a survey of the research on event boundary detection algorithms that help in identification of event regions, which is a basic building block of many spatial queries. The paper identifies new areas where the event region boundary detection can be of significant importance. The paper emphasizes that the research community has, not yet adequately addressed the problem of designing a general-purpose in-network distributed spatial query processor able to handle spatial queries. Some work has been done for in-network distributed boundary detection and spatial analysis, but there is still significant unexplored research space in this field.



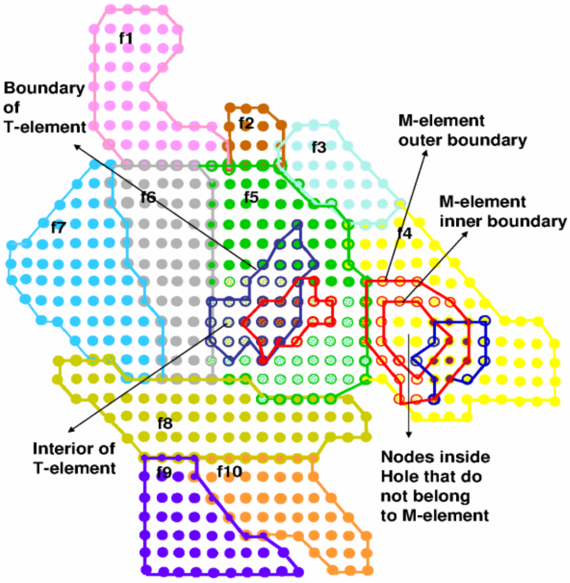
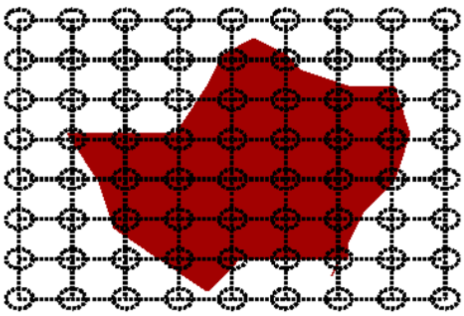


Figure 1(a):Fields (f1-f10) with event Geometries in a vineyard **(b) Example WSN over Figure 1(a)**



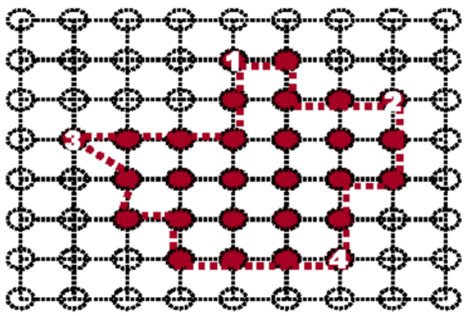


Figure 2(a):Event Geometry **(b) Sensed Event Geometry**

References

[1] J. K. Hart, K. Martinez, Environmental Sensor Networks: A revolution in the earth system science?, Earth-Science Reviews 78 (2006) 177-191. **doi:10.1016/j.earscirev.2006.05.001**

[2] T. Ulrich, Wireless network monitors H2O: System saves resources, increases yield in cabernet vineyard, Wines and Vines Magazine.

[3] F. J. Pierce, T. V. Elliott, "Regional and on-farm wireless sensor networks for agricultural systems in eastern washington," *Comput. Electron. Agric*. 61 (1) (2008) 32-43. **doi: 10.1016/j.compag.2007.05.007**.

[4] A. Baggio, "Wireless sensor networks in precision agriculture," in *Proc. of Workshop on Real-World Wireless Sensor Networks. REALWSN'05*, Stockholm, Sweden, 2005.

[5] J. Burrell, T. Brooke, R. "Beckwith, Vineyard computing: Sensor networks in agricultural production," *IEEE Pervasive Computing*, 3 (2004) 38-45. **doi: 10.1109/MPRV.2004.1269130**.

[6] J. Gehrke, S. Madden, "Query processing in sensor networks," *IEEE Pervasive computing*, 3 (2004) 46-55. **doi:10.1109/MPRV.2004.1269131**.

[7] G. J. Pottie, W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, 43 (5) (2000) 51-58. **doi:10.1145/332833.332838**.

[8] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, "TinyDB: an acquisitional query processing system for sensor networks.," *ACM Trans. Database Syst.*, 30 (1) (2005) 122-173. **doi :10.1145/1061318.1061322**.

[9] F. Jabeen, S. Nawaz, S. Tanveer, M. Iqbal, "Spatio-temporal query processing over sensor networks: Challenges, state of the art and future directions," *TIIS*, 6 (7) (2012) 1756-1776. **doi:10.3837%2Ftiis.2012.07.002.**

[10] Wireless sensing at camalie vineyard, http://www.crossbow.com (2013).

[11] I. Khan, H. Mokhtar. and M. Merabti, "A survey of boundary detection algorithms for sensor networks, " in *Proc. of the 9th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, UK, June 2008. **doi:10.1504/IJSNET. 2010.034065.**

[12] Kun-Ying Hsieh, Jang-Ping Sheu, "Hole detection and boundary recognition in wireless sensor networks," *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on* , vol., no., pp.72,76, 13-16 Sept. 2009. **doi: 10.1109/PIMRC.2009.5449912** .

[13] M. Fayed, H. T. Mouftah, "Localised convex hulls to identify boundary nodes in sensor networks," *Int. J. Sen. Netw*. 5 (2) (2009) 112-125. **doi:10.1504/IJSNET.2009.024681**.

[14] Chi Zhang, Yanchao Zhang, and Yuguang Fang. 2006. "Localized coverage boundary detection for wireless sensor networks." in *Proc. of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks* (QShine '06). ACM, New York, NY, USA, Article 12. **doi:10.1145/1185373.1185390** .

[15] W. Hu, N. Bulusu, C. T. Chou, S. Jha, A. Taylor, V. N. Tran, "Design and evaluation of a hybrid sensor network for cane toad monitoring," *ACM Trans. Sen. Netw*. 5 (2009) 4:1-4:28. **doi: 10.1145/1464420.1464424**.

[16] G. Tolle, J. Polastre, R. Szewczyk, D. E. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buon-adonna, D. Gay, W. Hong, "A macroscope in the redwoods," in *Proc. of SenSys*, 2005, pp. 51-63. **doi: 10.1145/1098918.1098925.**

[17] Q. Fang, J. Gao, L. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proc. of INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 4, March, pp. 2458-2468 vol.4. **doi:10.1109/INFCOM.2004.1354667**.

[18] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical Edge Detection: Learning and Evaluating Edge Cues," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 25, No. 1, pp 57-74, 2003. **doi:10.1109/TPAMI.2003.1159946**.

[19] J. C. Bezdek, R. Chandrasekhar, and Y. Attikiouzel, "A Geometric Approach to Edge Detection," *IEEE Transactions On Fuzzy Systems*, Vol. 6, No. 1, pp 52-75,1998. **doi:10.1109/91.660808**.

[20] Robert M. Haralick, "Edge and region analysis for digital image data," *Computer Graphics and Image Processing*, Volume 12, Issue 1, January 1980, Pages 60-73, ISSN 0146-664X, **doi:10.1016/0146-664X(80)90004-0**.

[21] Uvika, Sumeet Kaur, "A Comprehensive Review On Different Edge Detection Techniques," *An International Journal of Engineering Sciences*, ISSN: 2229-6913, Issue July 2012, Vol. 6, pp 163-169,

[22] Sonam Saluja, Aradhana Kumari Singh, Sonu Agrawal, "A Study of Edge-Detection Methods," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 1, January 2013, pp 994-999.

[23] K. Chintalapudi, R. Govindan, "Localized edge detection in sensor fields," *Ad Hoc Networks,* 1 (2-3) (2003) 273-291. **doi:10.1016/S1570-8705(03)00007-6**.

[24] J. S. Deogun, S. Das, H. S. Hamza, S. Goddard, "An algorithm for boundary discovery in wireless sensor networks," in *Proc. of HiPC*, 2005, pp. 343-352. **doi:10.1007/11602569_37**.

[25] S. Selvakennedy, "An energy efficient event processing algorithm for wireless sensor networks," in *Proc. of MSN*, 2006, pp. 588-599. http://link.springer.com/chapter/10.1007/11943952_50.

[26] G. Jin, S. Nittel, NED "An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks," in *Proc. of MDM*, 2006, p. 153. **doi:10.1109/MDM.2006.110.**

[27] M. Ding, D. Chen, K. Xing, X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proc. of INFOCOM, 2005*, pp. 902-913. **doi:10.1109/infcom.2005.1498320.**

[28] C. Zhang, Y. Zhang, Y. Fang, "Localized coverage boundary detection for wireless sensor networks," in *Proc. of the 3rd international conference on*

Quality of service in heterogeneous wired/wireless networks, QShine '06, ACM, New York, NY, USA, 2006. doi:10.1145/1185373.1185390.

[29] K. Ren, K. Zeng, W. Lou, "Fault-tolerant event boundary detection in wireless sensor networks," in *Proc. of GLOBECOM*, 2006. doi: 10.1109/GLOCOM.2006.514.

[30] B. Krishnamachari, S. S. Iyengar, "Efficient and fault-tolerant feature extraction in wireless sensor networks," in *Proc. of Zhao and Guibas*, pp. 488-501. http://link.springer.com/chapter/10.1007/3-540-36978-3_33#page-1.

[31] B. Krishnamachari, S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event re-gion detection in wireless sensor networks," *IEEE Trans. Comput.*, 53 (3) (2004) 241-250. doi: 10.1109/TC.2004.1261832.

[32] P.-K. Liao, M.-K. Chang, C.-C. Kuo, "Statistical edge detection with distributed sensors under the neynan-pearson (np) optimality," in *Proc. of Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, Vol. 3, May, pp. 1038-1042. doi:10.1109/VETECS.2006.1682992.

[33] D. Devaguptapu, B. Krishnamachari, "Applications of localized image processing techniques in wireless sensor networks," in *Proc. of SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, Florida, April 2003. *DOI: 10.1117/12.487490*.

[34] Y. Wang, J. Gao, J. S. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proc. of the 12th annual international conference on Mobile computing and networking, MobiCom '06, ACM*, New York, NY, USA, 2006, pp. 122-133. doi:10.1145/1161089.1161104.

[35] R. Nowak, U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *Proc. of Zhao and Guibas* [38], pp. 80-95. doi:10.1007/3-540-36978-3_6.

[36] Y. Liao, M. Qi, "Boundary discovery based on overlapping clustering in non-uniform wireless sensor networks," in *Proc. of Information Science and Applications (ICISA), 2011 International Conference on*, April, pp. 1-7. doi:10.1109/ICISA.2011.5772380.

[37] A. Jaffer, M. Jaseemuddin, M. Jafarian, H. El-Sayed, "Event boundary detection using autonomous agents in a sensor network," in *Proc. of AICCSA, 2007*, pp. 217-224. doi:10.1109/AICCSA.2007.370886.

[38] F. Jabeen, A. A. A. Fernandes, A. A. A. Fernandes, "Distributed spatial analysis in wireless sensor networks," in *Proc. of ICPADS, 2010*, pp. 558-567. doi:10.1109/ICPADS.2010.58.

[39] A. Singh, M. J. Stealey, V. Chen, W. J. Kaiser, M. Batalin, Y. Lam, B. Zhang, A. Dhariwal, C. Oberg, A. Pereira, G. S. Sukhatme, B. Stau er, S. Moorthi, D. Caron, M. Hansen, "Human Assists Robotic Team Campaigns for Aquatic Monitoring," *Journal of Field Robotics*, 24 (11) (2007) 969-989. doi:10.1002/rob.20225.

[40] Galpin, C. Y. A. Brenninkmeijer, F. Jabeen, A. A. A. Fernandes, N. W. Paton, "Comprehensive optimization of declarative sensor network queries," in *Proc. of SSDBM, 2009*, pp. 339-360. doi:10.1007/978-3-642-02279-1_26.

[41] M. Schneider, "Spatial Data Types for Database Systems: Finite Resolution Geometry for Geographic Information Systems," Vol. 1288 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 1997. doi:10.1007/BFb0028319.

[42] F. Jabeen, A. A. A. Fernandes, A. A. A. Fernandes, "An algorithmic strategy for in-network distributed spatial analysis in wireless sensor networks.," *Journal of Parallel and distributed computing*, 2012, pp. 1628-1653. doi:10.1016/j.jpdc.2012.09.005.

[43] N. P. N. D. T. Grifths, A.A.A. Fernandes, "Realm-based temporal data types.," in *Proc. of Preprint Series*, Department of Computer Science, University of Manchester, 2002.

[44] L. Mottola, G. P. Picco, "Logical neighborhoods: A programming abstraction for wireless sensor networks," in *Proc. of DCOSS*, 2006, pp. 150-168. doi:10.1007/11776178_10.