Spatio-Temporal Query Processing Over Sensor Networks: Challenges, State Of The Art And Future Directions

Farhana Jabeen[1], Sarfraz Nawaz[2], Sadaf Tanveer[1] and Majid Iqbal[1]
[1] COMSATS University of Science and Technology Islamabad - Pakistan
[2] Computer Laboratory, University of Cambridge, CB3 0FD, United Kingdom
[e-mail: {farhanakhan,sadaf, majid_iqbal]@comsats.edu.pk , sarfraz.nawaz@cl.cam.ac.uk]
*Corresponding author: Farhana Jabeen

Received April 4, 2012; revised June 6, 2012; accepted June 21, 2012; published July 25, 2012

Abstract

Wireless sensor networks (WSNs) are likely to be more prevalent as their cost-effectiveness improves. The spectrum of applications for WSNs spans multiple domains. In environmental sciences, in particular, they are on the way to become an essential technology for monitoring the natural environment and the dynamic behavior of transient physical phenomena over space. Existing sensor network query processors (SNQPs) have also demonstrated that in-network processing is an effective and efficient means of interaction with WSNs for performing queries over live data. Inspired by these findings, this paper investigates the question as to whether spatio-temporal and historical analysis can be carried over WSNs using distributed query-processing techniques. The emphasis of this work is on the spatial, temporal and historical aspects of sensed data, which are not adequately addressed in existing SNQPs. This paper surveys the novel approaches of storing the data and execution of spatio-temporal and historical queries. We introduce the challenges and opportunities of research in the field of in-network storage and in-network spatio-temporal query processing as well as illustrate the current status of research in this field. We also present new areas where the spatio-temporal and historical query processing can be of significant importance.

Keywords: WSN, flash memory, in-network processing, spatial queries, historical queries, persistent storage, spatio-temporal query Processing

http://dx.doi.org/10.3837/tiis.2012.07.002

1. Introduction

A *wireless sensor network* (WSN) may be viewed as a distributed computing platform [1], with each node as computational resource and not just a data collection and data transmission device. Albeit limited, node resources such as processing power and memory can be used to execute application logic. WSN can be construed as an intelligent and largely autonomous instrument for scientific observation at fine temporal and spatial granularities over large areas. Each value measured by a sensing device integrated with a particular node can have associated with it the location of the sensor node at the time of measurement, the time at which measurement was taken and the nature of the measured reading. The spectrum of applications for WSNs spans multiple domains. In the environmental sciences, in particular, they are on the way to become an essential tool for monitoring complex physical phenomena. This is testified by many reported deployments, for example, close to fifty environmental sensor networks were surveyed in [2]. The importance of identifying, tracking and reporting relationships between dynamic, and transient spatial phenomena and application-specific geometries has been stressed in environmental monitoring applications. As an example of the usefulness of this kind of 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 [3], and is interested in part of a field where the soil moisture has dropped below a certain threshold, so that only those parts are irrigated, given the limited water supply. The sensor nodes sense the soil moisture and, using their short-range radio, communicate with each other to send the real-time information from the fields to the farmer. Such a WSN, therefore, allows the farmer to get a real-time digital picture (in the form of sensed measurements) of the physical world. The raw data being collected, enables the farmer to see what is going on in the fields and to adjust his management strategies. In environment monitoring applications, networks are deployed with the aim of long-term monitoring in the field. Data analysis in such applications includes searching for patterns or events, feature extraction, topological relationships of physical phenomena with application-specific geometry (e.g., whether the low moisture event region is adjacent, or inside, or outside a cultivated field).

Because of resource constraints, it is not energy-efficient for each node to send every sensed measurement towards some external repository for storage, where the required data analyses can be performed off-line. One alternative that may extend the network life is to store the data and events in the sensor nodes themselves, using persistent storage devices such as flash memory units and then to push queries on to the network and into the node. But, this approach involves addressing challenges like storage limitations, spatio-temporal query language, spatio-temporal, and historical algebra that make it difficult to implement in practice. In-network processing uses distributed algorithms for information processing with a view, primarily, of reducing network traffic and thus improving the longevity of deployments at larger scales. TinyDB [4], Cougar [5], and SNEE [6] are some of the examples of available in-network declarative query processing over WSNs. These pieces of work have shown that a WSN can be viewed as a distributed database. This has led to the approach of retrieving data from a WSN by viewing it as the computational environment upon which SQL-like declarative queries are compiled and optimized to run. Declarative queries allow users to specify only what data they want from a WSN. Programming WSNs requires specialized knowledge, the scarcity of the resources puts a tight limit on code size, and debugging is cumbersome. Declarative queries on the other hand, allow for low-cost repurposing, since rather than cumbersome reprograming of the network, users can just pose a different query to the SNQP [4]. However, current SNQP's are unable to cater for a wide range of sensor network applications because they do not take spatio-temporal requirements into account. Existing declarative query languages for WSNs provide support for obtaining and performing simple analyses on live data. However, the support for situating data in time or in space and spatio-temporal query-level support is non-existent. Another aspect that is missing from current SNQPs is the analysis of historical data. Access to live data is helpful in real-time monitoring but sometimes it is required to perform analysis on historical data. This requires a storage manager to handle decisions as to how and where to store data so that it can be accessed in an energy-efficient manner with low latency.

There are many environmental monitoring applications (such as oil spills, habitant monitoring [7], precision agriculture [8] or forest fires) that require spatial and temporal extensions to query language and the algebra. The challenge is to extend sensor network query processors to include the support for spatio-temporal data retrieval and analysis. Sensor network applications can be divided into two broad areas, viz., data-driven and event-driven. In data-driven applications, scientists are interested in collecting measurements to perform detailed analysis; of continuous time-series of observations. In a sensor network, each sensing device on a node can generate a data stream into which tuples are placed often at well-defined time intervals and in fixed order. These tuples normally stay for a limited time period in the memory of the sensor node. However, if the same reading is required for later reference, then one may need to store it explicitly locally in flash memory. This is made even more challenging by limited storage available and the fact that storage management is expensive.

In event-driven WSN applications (e.g., precision viticulture [3], volcano monitoring [9]) scientists are interested in the shape and size of an event as it occurs in the sensing scope of the WSN. The event geometry, which we refer to as an induced geometry, is definable in terms of the location in space of those sensor nodes that satisfy the event predicate (e.g., humidity>98 and temperature<10). Continuous monitoring of event geometries makes it possible to track the spatial evolution of the underlying spatial phenomenon. For supporting evaluation of spatio-temporal historical queries, it is necessary to store the information related to such geometries. This mechanism will allow users to keep track of the changing state of an object or of those properties for which a history is maintained. This is done by tracking the changes that result from assignments made to a property or relationship over time and identifies such occurrence of change with a timestamp and the corresponding value. As sensor nodes are equipped with limited persistent storage, there would be a need that each node be equipped with a sophisticated storage manager for supporting historical spatio-temporal queries.

The main contributions of this paper are as follows:

- We have presented few motivating environmental monitoring applications where spatio-temporal query processing can be beneficial. Based on the motivating example, we have also identified some of the spatial, temporal, spatio-temporal queries, the in-network evaluation of which can be beneficial for such applications.
- We have outlined the challenges and opportunities of in-network spatio-temporal query processing in WSNs and also the features that should be supported by storage managers.
- We have surveyed the work related to spatial/temporal analysis, storage managers and management models for evaluation of spatio-temporal and historical queries in WSNs.
- Finally, we have depicted future research scopes of spatio-temporal query processing in WSNs.

The remainder of the paper is organized as follows: Section 2 describes how in-network spatio-temporal query processing in environmental monitoring application can be more beneficial. In this section some of the example spatial, temporal, spatio-temporal, spatio-tempral historical queries are given. Section 3, highlights some of the challenges in supporting in-network execution of spatio-temporal and historical queries. Section 4, survey the work related to supporting spatial/temporal analysis, storage managers and management models for evaluation of spatio-temporal and historical queries. Section 5, focuses on some discussions that include available solutions, future research scope, open issues and finally, Section 6, concludes the paper.

2. Motivation

A number of recent deployments of sensors have been made for environmental monitoring purposes: about 25 nodes were deployed in Camalie vineyards [3], 45 sensor nodes in *Network Avanzato per il Vigneto* (NAV) system [10] for precision viticulture, 150 nodes in LofarAgro project [8] to fight fungal-disease in the field, about 26 nodes in AgFrostNet project [11] for frost monitoring, and 200 nodes in Great Duck Island project [7], to monitor burrow occupancy and the environmental changes occurring inside the burrow and on the surface during the period of the breeding season. In most of these environmental monitoring applications, the WSN is, by and large, an isolated system with depletable resources. Replacement of the batteries of sensor nodes under such scenairios is not feasible because of the following reasons: (1) time consuming and expensive; (2) minimal disturbance is crucial for avoiding distortion in results [7] (the birds under study might change their behavioural patterns or distributions).

WSNs have been used in precision viticulture for suggesting appropriate management practices to increase productivity and the quality of the crops [3][8][10][11]. 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). The regular spray of the pesticides needs to be stopped in high temperature or low moisture, as it is dangerous for the health of the crop. Similarly, climate poses a major threat of plant injury due to low-temperature. For frost protection, some wine grape growers rely primarily on wind machines, while others, use wind machines and irrigation in combination to adequately modify temperature. These frost/freeze protection systems are expensive. Therefore, such measures should be taken only if the air temperature reaches a critical threshold. Real-time temperature monitoring combined with historical data insight could prevent frost effects. The supply of nutrients to the plants depends highly on soil moisture. The pH is a measure of the intensity of the acidity. Wines with low to moderate pH tend to be crisper and age better as compared to wines with higher pH levels.

In all these scenairios, nodes collect temperature, moisture, and other environmental properties. Every node transmits sensed value to some destination that is external to the WSN for storage and off-line analysis, that may be prohibitively expensive and sometimes not possible, given the typical data collection rates and network sizes. This approach is also referred as warehousing. In this approach, apart from network longevity, scalability is an issue as it will result in increased bandwidth requirements, raising the risks of packet loss. In addition, the nodes that lie closer to the sink consume energy at a much faster rate, as they have to relay more packets as compared to those nodes that lie at the far edge of the network. This can significantly reduce the scalibility and lifetime of large sensor networks. In the last decade, scalable MAC and routing protocols for sensor networks have been well addressed, but the scalability of in-network storage and in-network spatio-temporal analysis in a cost-effective manner has been largely overlooked.

Spatio-temporal queries will enable us in identifying problematic areas or areas prone to disease and will allow us to analyze and monitor the temperature and soil moisture with temporal and spatial precision in order to decide when and where to apply the chemicals, and when to start or stop the protective systems. Such queries will also enable us to create a humidity map, to analyze the spatial variations of soil and nutrients and to create a map of soil vigour potential to determine why the soil vigour is too small or too large. Grape growers take advantage of natural factors, by selecting suitable sites for particular grape varieties. Variations in the physical conditions and climatic enviroment usually result in vine growth, the quality of which is not uniform throughout the vineyard. Wine makers achieve greater control over the product by selecting, fermenting and blending batches of grapes with suitable ripeness and flavors. Batch selection could be decided with the assistance of spatio-historical analysis of data related to grapes in the vineyard and environmental factors affecting their development.

Historical analysis allows for the traceability of product through a record. It will allow the farmers to map the physical conditions and climatic environment with the quality of the yield. The growers will be able to see that how much spatial variation has occured in crop yield in response to management practices under varying natural conditions. This could influence future management decisions and will allow to apply a variable approach to each vineyard block or field. Spatio-temporal query processing will allow the farmers to determine spatial variations in vegetation balance, vigour, health and soil moisture etc., and to correlate the aforementioned factors with maps of soils, microclimate, topography etc.

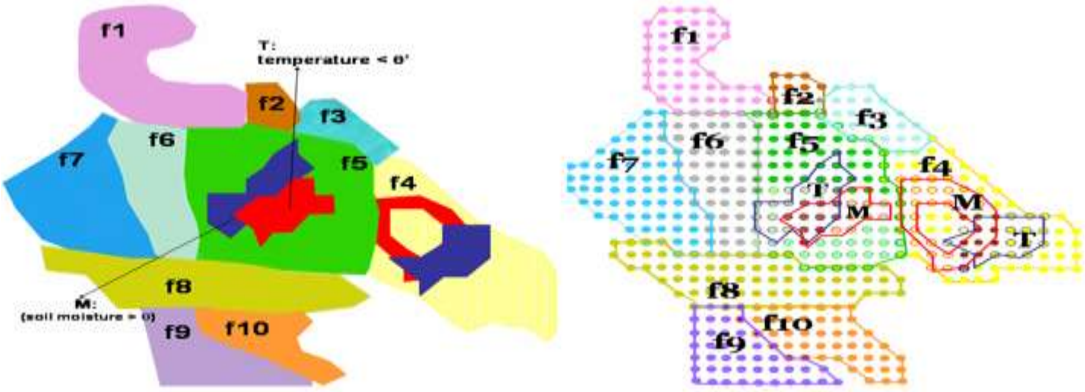


Fig. 1. (a) Fields (f1-f10) with event geometries in a vineyard (b) Example WSN over (a) showing approximate geometry membership

Based on the precision viticulture application, this section will present some examples of spatial, temporal, spatio-temporal, and spatio-temporal historical queries. One real-world example in which WSNs are being deployed for precision viticulture is at Camalie Vineyards [3]. We have used the underlying permanent geometries (***f 1-f10***) in this deployment (shown in **Fig. 1.**) as the basis for our examples (See http://camalie.com/CamalieGIS/Naked/default.asp [12 June 2011] for the geometries we use as our basis.). Permanent geometries (which we refer to as an asserted geometry) are representations of physical features (e.g., a well or a cultivated field). **Fig. 1.**, shows the asserted (***f 1-f10***) and induced (***M*** and ***T)*** *geometries*. More formally, given thresholds ***X*** and ***X'***, let ***M*** be the induced geometry where soil moisture is above ***X*** and ***T*** the induced geometry where temperature is below ***X'***. Both geometries ***M*** and ***T*** have two elements. Both elements of geometry ***T*** are without holes. One of the elements of geometry ***M*** has a hole and other is without one. Note that the nodes within a hole do not belong to the interior of the corresponding geometry. An asserted geometry of type region may contain holes such as a cultivated field with a hole representing where a lake is located and so on.

2.1 Spatial Queries

- Every 20 minutes, report the average temperature from the geometry ***M*** lying vertex inside ***f5.***
- Report whether there is a need to spray pesticides in ***f5***. (Note: There will be a need to spray pesticides, if the query ((***M*** intersects ***f5***) and ***((T*** intersects ***f5***) **and** (***M*** intersects ***T***))) returns True)
- Report the area of the region lying at the intersection of ***f5***, ***M*** and ***T*** . (i.e., to evaluate the query containing the following expression area((***M*** intersection ***f5***) and ((***T*** intersection ***f5***) and (***M*** intersection ***T***)))
- Report the largest area among the M geometries in f5 plus ***f6***.
- Every 20 minutes, report the maximum temperature from ***f5*** which is not lying at the intersection of geometry ***M*** and ***T*** . (i.e., f5 minus (T intersection ***M***)).
- Every 10 minutes, report the average temperature from sensor nodes, which are lying at distance greater than ***5 meters*** from geometry ***M*** .

2.2 Temporal Queries

- Every 30 minutes, report the stream of temperatures between 1:00 am and 9:00 am today.
- Report the time, when the induced geometry ***M*** is first detected today.
- Every 30 minutes, report the stream of time and locations where above-ground temperature is greater than temperature threshold between 7:00 am and 9:00 am.
- Report the stream of maximum and minimum pH-levels for each day, between 1:00 pm and 5:00 pm, during the previous month.
- Report the stream of time and locations, where the temperature is below 0 degrees between 1:00 am and 10:30 am during the previous week.

2.3 Spatio-Temporal/ Spatio-Temporal Historical Queries

- Every 60 minutes, report the minimum temperature from sensors areainside field ***f5*** minus the ones that are lying within 5 meters distance from ***T*** .
- Every 60 minutes, obtain a stream of average, maximum and minimum temperatures in the last 60 minutes of every temperature source areainside ***f5***.
- Every 10 minutes, report times in the last hour when the average temperature in field ***f5*** was the same as the maximum temperature in the intersection of its boundary with river.
- Report three times, when highest average temperature intersects field ***f5*** recorded in the previous week.
- Every 24 hours, obtain a stream of maximum and minimum temperatures for the past 7 days from every temperature source areainside ***f5***.
- Every 30 minutes, report the stream of maximum and minimum per day pH-levels, for the past 01 month from every temperature source areainside ***f5***.
- Report previous day's minimum and maximum temperature, moisture, and pH-levels from every source areainside (***f5*** plus ***f4*** plus ***f6)***.
- Every 30 minutes, report hourly average of the previous day's temperature, pH-level and moisture from every source areainside vineyard.

3. Challenges

Spatio-temporal querying over sensor networks clearly raises research issues at several levels. Some of the challenges are as follows:

3.1 Persistent Storage

For persistent storage, sensor nodes rely on flash memory. Despite the advances in flash technologies, a sensor network cannot anticipate to emulate the volume and size of classical devices available at a base station outside the sensor network. Flash memory is a specific type of *Electrically Erasable Programmable Read-Only Memory* (EEPROM). A sensor network still cannot handle the storage and querying requirements of monitoring applications that depend on very long time-series. **Table 1** discusses the storage capacity of different commercially available sensor nodes [12]. Flash memory has a number of distinct characteristics as compared to other storage media that make efficient storage management a challenging task. A flash memory is usually organized in blocks that consist of pages. In flash memory, data cannot be over-written due to the physical erase-before-write characteristics. Only single or multiple pages of memory can be erased at a time. Each page can be written only a limited number of times. The number of erase operations allowed to each block is also limited.

Table 1. Storage capacity of sensor nodes

	Mica2	**Micaz**	**Imote 2**	**IRIS**	**TelosB**	**Tmote Sky**
Data Storage	512KB	512KB	32MB	512KB	1MB	1MB

In-network storage must rely on policies and strategies to optimize the use of scarce space with a view to supporting queries over the widest possible range, in time and space, of values. It may be advantageous to support an aging policy (that is the more recent the data values, the less summarized they are) as a result of which older data is stored in coarser-grained summaries, implying that the queries posted over the old data will be responded with much less precision. But at the same time, storing the data for future reference gives rise to several compromises. For example, how good an approximate answer should be, given the trade-off between accuracy and level of summarization, for how long should the data be stored, when should aging policies be applied, how to trade-off compression benefit and query performance, how to trade-off the availability of data for applications and network longevity, etc. Using compressed summaries will also lead to access mixed methods as some data will be transformed at higher resolution, other at lower resolution and some not at all. The greater the number of access methods, higher will be the computation requirements for compression and decompression of the data as well as for merging the results produced by various access methods. For most of the historical queries, an efficient search strategy is also required. Sometimes, it is possible that the measurements are collected and stored in the history at one time granularity but afterwards it is required to report the queries responses based on the recorded data at different time granularity.

3.2 Distributed Data

WSN is a distributed platform, therefore, each node is only aware of the part of the induced geometries that lie within its sensing range and that part of asserted geometries that lie within its radio range. A query related to the detection of an induced geometry is re-evaluated with some periodicity and each node independently updates the local information that defines the induced geometries it is a member of. The inherent scarcity of resources and the nature of underlying platform, where execution is distributed and carried out periodically over sensed data streams, gives rise to non-trivial challenges of defining framework for spatio-temporal analysis, identification of spatial, temporal and historical abstract data types, to define the operations that can be supported by the spatial, temporal and historical data types and to design algorithms for these operators over the geometries that can be represented over the framework, with the optimization goal of energy, storage efficiency and short response times. The resolution or scale of the spatial data may vary, geometries may have different spatial dimensions, and spatial types (e.g., point, line, or region). Following the algebraic approach [13,14,15], application-specific geometries can be defined based on abstract data types such as *point* (a value of which could denote, e.g., a well), *line* (a value of which could denote, e.g., a river, or a pipeline), *region* (a value of which could denote, e.g., a building, or a cultivated field). These several forms of diversity give rise to challenges on how to integrate and to keep them consistent in order to provide correct answers for queries.

3.3 Aggregation of Information

Recall that information about a geometry lies in a distributed fashion inside the WSN and that each node is only aware of that portion of induced geometry that is in its sensing range and that part of asserted geometry that is in its communication range. This implies that the computation of spatial operations requires aggregating information from all the nodes that belongs to the operands of the operators involved. An operand may comprise a single-element geometry or a multi-element geometry with or without holes. Furthermore, induced geometries are dynamic, which implies that, their shape and size cannot be known in advance. This requires the design of efficient in-network hierarchial aggregation scheme that allows for aggregation of information, from single and multi-element geometries with or without holes to compute the final result. Over the top of spatial aggregation, it would be very challenging to design some technique to perform in-network aggregation functions like min, max, average over the sensed measurements and temporal aggregation functions like per hour. For example, consider the query "*Every 60 minutes, obtain a stream of* ***average, maximum and minimum*** *temperatures in the* ***last 60 minutes*** *of every temperature source* ***areainside*** *f5"*. Let us suppose that the temperature is sensed every 05 minutes.

3.4 Synchronization in Complex Queries.

For timely coordination, participating entities must start and finish the processing of each spatial, temporal and historical operator in the query at the appropriate time in order to avoid delays in response and in achieving accuracy. For example in the evaluation of a spatial query which may comprises several spatial operators, the nodes can only participate in the evaluation of spatial operator, if they are part of one or both operands. During the evaluation of a spatial query, at each point in time, there may be situations where some of the nodes satisfy the participation requirements of one or more spatial operators in the query, and some satisfy for other operators, and others do not satisfy for any of the operator. For example, upon in-network evaluation of the query satisfying algebraic expression ((***M*** intersects ***f5***) and ***((T*** intersects ***f5***) **and** (***M*** intersects ***T***)))), it is possible that some nodes are part of geometry **f5**, **M** and **T**, other part of **f5** only, and some do not satisfy any of the operand etc.

3.5 Designing of a Spatio-temporal Query Language

To design a spatio-temporal query language which must include features to meet the functionality requirement of stream data in the light of real world deployed environmental sensor network applications. It would require development of operational semantics for the query language including typing and translation rules. As previously described, current sensor network query languages have no support for spatial, temporal and historical constructs (e.g., types and operations). The challenge is how to represent spatio-temporal types operations and then to integrate the resulting algebras into language. The basic motivation behind the declarative query approach is that of delegating to the system (i.e., spatio-temporal SNQP), the task of making the decisions that are needed to generate an optimized execution plan, specifying from which sensor nodes and which sensing devices to acquire data (or from which flash memory storage in which node to retrieve data), where and what to store, in which nodes and in which order to perform various database operations, when and where should intermediate results be sent and others. The system must strive to generate a query evaluation plan that not only meets the functional but also non-functional QoS expectations (such as response time, longevity of the network, result accuracy and others).

3.6 Query Evaluation Engine

One of the difficult challenges is the design of in-network spatio-temporal query evaluation engine with which each node can be equipped with for the evaluation of complex spatio-temporal queries using the corresponding operator algorithms. The algorithmic strategy of the evaluation engine for in-network distributed spatio-temporal analysis over WSN should be specifically tailored for energy-efficient in-network execution. The algorithmic strategy for the evaluation of complex spatial, spatio-temporal queries should be divided into logically-cohesive components thereby facilitating component reuse and sharing. Each node equipped with the evaluation engine must be allowed to participate in query dissemination, to contribute in the distributed evaluation of queries, to participate in the aggregation of intermediate results and in the routing of results to the user.

3.7 Storage Manager Features

By considering the various environmental monitoring applications described in the previous section various features/requirements that are envisaged for a storage manager are as follows:

- It should allow for writing data to multiple materialization points/files at a time. As a result, it would be possible to store the responses of the query involving sensing from different sensing attributes in separate materialization points. The query e.g., ***Materialize temperature, moisture acquired every 60 seconds***, involves two sensing attributes that are collected at the same acquisition rate and with same timestamps. It will be useful to store the pressure and temperature in separate files for future detailed analysis of individual sensing attribute as well as of both attributes in comparison.
- It should allow both reading and writing operation on an open file i.e. to utilize materialization point even if it has not been created in its entirety. This will allow to continuously perform analysis on the data as it is acquired, instead of waiting for the duration of the materialization query to complete. Forexample, in case of TinyDB [4] which makes use of Matchbox [16] storage manager, the materialization point, cannot be utilized until it is created in its entirety.
- It should allow for reading data from multiple files at a time. This will not only allow to perform comparison operation over the two materialized files e.g., to reply the query

Report the earliest time, location at which the induced geometry M is first detected today, but also to perform database operations such as SQL-project, and SQL-join.

- It should provide device life efficiency by supporting efficient wear-levelling mechanism. Updation of even a single byte in any page requires an expensive erase operation on the corresponding block before the new data can be rewritten. As each page on flash memory has limited life, flash file systems should use sophisticated data structures and algorithms, which allow for efficient not-in-place updates of data, reduce the number of erasures, and level the wear of the pages in the device.
- It should support high performance and reliability. For this purpose rollback schemes can be used.
- It must provide in-network data life efficiency, i.e., instead of erasing the data to generate space for new data, efficient aging policies must be defined that ellaborate what should be the levels of resolutions (i.e., how much compressed summaries).
- It must provide storage efficiency by figuring out how to store data, so that a complete page is not wasted, if only few bytes needs to be stored.
- It should support efficient searching mechanism to support spatio-temporal queries. These searching mechanism must allow to efficiently search data that is stored at different time granularity or because of aging policy have been summarized.
- Additional features includes: (1) support for files deletion, (2) updating information and metadata once stored.

4. Literature Review

At present, in-network spatio-temporal analysis in WSN is not catered for by a comprehensive, expressive, and well-founded framework. Spatio-temporal querying over sensor networks clearly raises research issues at several levels. It is crucial, therefore, to survey the state of art on several areas of the literature, as they may contribute insights, methods and techniques for building a complete framework for spatio-temporal query processing in sensor network.

4.1 Spatial Analysis

Farhana et al. [13][14][15] proposed the first generic framework capable of underpinning an algebraic approach to spatial analysis in WSNs as a distributed platform for in-network processing. This framework considers three kinds of geometry; viz., asserted, induced and derived. Derived geometries are obtained by applying spatial-valued operations to existing spatial values (e.g., applying intersection to two values of type regions would result in a new geometry of type, say, regions). The authors presented the definition of spatial algebra over the geometries representable by the framework. The spatial algebra comprises three spatial types, viz., point, line, and region. The value of spatial type can be single-element or multi-element. A value of type region may or may not have a hole. The authors have presented distributed in-network algorithms for the operations in the spatial algebra over the representable geometries, thereby enabling (i) new geometries to be derived from induced and asserted ones and (ii) topological relationships between geometries to be identified. The algorithms are specifically tailored for power-efficient in-network execution, with a focus on minimizing unnecessary communication and reducing the size of information to be communicated. This work suffers from the fact that it is not integrated with any existing WSN query processors.

The work described in [18] provides a computational model for WSNs to detect topological change (i.e., hole formation, hole disappearance, event region splitting, and event region merging) in dynamic regions based on local low-level snapshots of spatio-temporal data. The descriptions of the change is computed by the comparison of the event state at the consecutive two evaluation periods. At each evaluation period, once the boundary nodes are identified, the next step is the formation of groups based on the boundary nodes that have the same state in consecutive evaluation periods. The boundary nodes collect group information from their neighbours and participate in aggregation. After the reception of information from the group leaders, the sink node constructs the snapshot of the event region and the location of topological change. Farah and Zhong [19], propose an event-driven approach for capturing topological changes. The basic data structure used for the detection of topological changes is neighborhood ring. This work allows for the in-network detection of topological changes, but does not focus on reporting back the information related to topological changes. Jiang and Worboys [20] propose in-network algorithms for the detection and reporting of topological changes. The work allows for reporting the *Minimum Bounding Rectangle* (MBR) of the area where change has taken place and the type of topological change.

4.2 Management Models for Supporting Spatio-temporal/historical Queries

Five management models that have been used for querying spatial, temporal or historical sensor data are as follows:

4.2.1 Warehouse Approach.

With reference to Section 2, this model focuses on extracting each and every measurement from the sensor nodes and transmitting it for storage in some external repository, where the required data analysis is performed, off-line. We have already discussed in Section 2, that this approach may be very expensive. For the reduction in communication cost, there exist work which focus on computing the event and boundary of the event geometry [21] in a distributed manner using in-network processing techniques. The aim is to transmit the information only from the boundary nodes of the event geometry towards sink. Sink is held responsible for creating snapshot of the event geometry from the boundary information. This work related to inducing event geometry can act as a stepping stone to achieve our broader goal of cost-effectively performing in-network spatio-temporal analysis.

By using the geometries shown in **Fig. 1**, we have conducted an experiment using warehousing approach and compared the results with our experiment related to sending only event information from boundary nodes towards the sink. The trade-off is between processing costs against transmission costs. The aim of the experiment is to show that, particularly for multi-hop networks, performing in-network processing is more energy-efficient.

We have implemented the algorithm in nesC over TinyOS [22] and used PowerTOSSIM [23] for power modeling. The specification of the sensor nodes we have simulated is [Type = MICA2, Radio = CC1000, Energy Stock = 31,320,000 mJ (2 Lithium AA batteries)]. The radio range is set such that the nodes can form a one-hop neighbourhood. The radio connectivity between nodes is based on distances between nodes and, therefore, the maximum cardinality of the one-hop neighbourhood of a node is set to eight in our experiments.

It is assumed that each node knows its parent, to which it has to transmit message, and, to receive messages from, its children. A node tries to send a packet to its parent up to four times if it does not receive an acknowledgement from parent. Otherwise, it broadcasts the message to its neighbours. Nodes also make sure to not forward duplicate packets. The results of these simulations are shown in **Fig. 2**. In this experiment, for a network size of 166 nodes, the amount of energy consumed by CPU and radio is close to 2,573 kmJ. If each node is powered by two AA batteries, the initial energy stock per node is 31,320,000 mJ. The total energy stock inside the network is then the network size times 31,320,000. This implies that each evaluation episode consumes between 0.05% and 0.09% of the total energy stock for networks containing between 166 and 223 nodes, respectively. It can be seen that cost of sending event information from each node part of a network is much higher as compared to our experiment in [13][14][15], in which we only transmit the event information from boundary nodes. It can be seen in [13][14][15], that each evaluation episode consumes between 0.008% and 0.016% of the total energy stock for networks containing 166 and 223 nodes. This is low enough that adding the energy required to induce the event geometries (not counted in [13][14][15]) is unlikely to significantly detract from the force of these conclusions.

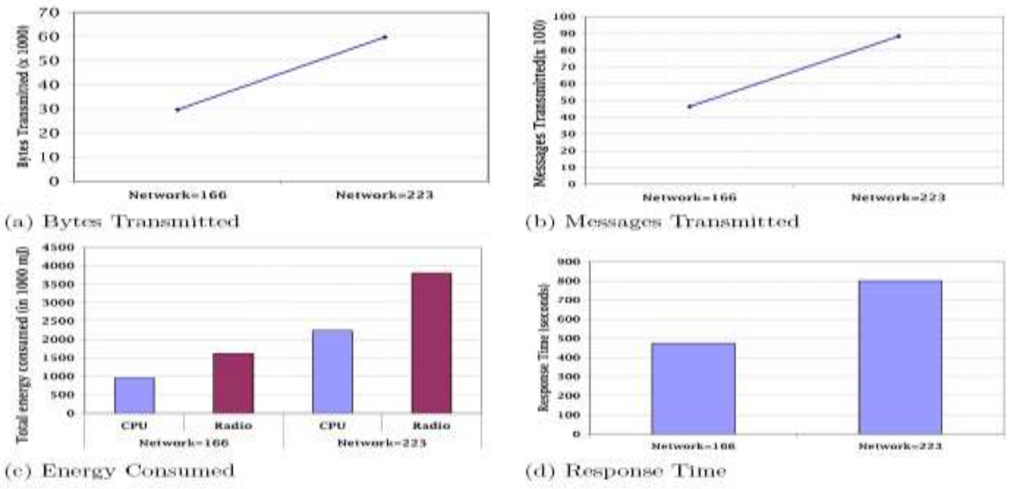
4.2.2 Suppression Approach

Suppression refers to techniques that help in reducing the cost of reporting changes in sensor values. These techniques are generally independent of queries. Temporal suppression refers to a decision not to report a value from a node if, that value has not changed since the last time it was reported from that node. This kind of suppression aims for a reduction in communication, but it cannot contend with a situation in which e.g., the temperature changes across an entire region to the same value. Spatial suppression techniques avoid simultaneous transmission of identical values by a group of neighbours. One of the main challenges faced in suppression techniques, is that lost messages must be distinguished from the suppression event.

In [24], the authors contribute a spatio-temporal suppression technique, called constraint chaining (CONCH), for minimizing energy cost in applications where change is slow or predictable and data is spatially correlated.

4.2.3 Query data using Aggregation Schemes

The third model focuses on in-network aggregation of the data produced by the sensor nodes. Examples include averages [25], minima and maxima [26], histograms [27] and summaries based on wavelets or distributed regression [28]. Although, this approach results in better energy efficiency than the first because only the aggregates are transmitted, it fails to retain the shape and structure of the data and only provides aggregate statistics. This approach is also not effective for spatio-temporal data analysis as the temporal and spatial aspects of data are lost.



(a) Bytes Transmitted (b) Messages Transmitted

(c) Energy Consumed (d) Response Time

Fig. 2. Experiment O1: Behaviour w.r.t. Network Size

4.2.4 Query data using Approximation Schemes

The fourth model addresses the need to query historical data by using approximation as opposed to aggregation techniques to reduce the communication. These approximation techniques are based on discovering and exploiting statistical models to report approximate answers probabilistic error guarantees [29]. Various models have been proposed according to which either the sensor nodes only push unexpected results or the base station only pulls data from a specific node when the query posted by the user requires greater accuracy than provided by the model at the base station. Such approaches optimize the collection of data at the cost of data correctness. However, they cannot replace individual readings and incur a greater risk of misunderstanding when the environment is not yet fully understood.

The authors focus on incorporating long-term storage and search for spatio-temporal patterns in sensor data collected for data-intensive scientific applications [30]. The approach is based on constructing and storing multi-resolution summaries of data using wavelet transforms. Queries requiring a lowest-precision response are processed locally at the root; otherwise it is passed down the hierarchy to the nodes containing the relevant data at higher resolutions. To ameliorate the problem of limited storage, a progressive data aging strategy is used. DIMENSIONS approach also uses a multi-resolution hierarchical index that enables it to efficiently answer queries by hashing to the appropriate nodes. However, it does not support spatial queries such as GetNearest as it is not designed for storing location information. Many interesting challenges remain such as designing load balancing schemes to avoid the root of the hierarchy becoming a bottleneck, obtaining energy efficiency through compression, negotiating trade-off between compression benefit and query performance, distribution of processing at different levels of hierarchy and choosing good aging strategies.

In [31] the authors consider a push-pull approach for executing one-shot precision-constrained approximate queries. They propose a generic two-tier data storage strategy where the sensor nodes store high-precision data and the base station stores approximation only. The authors also describe node selection techniques for use in refresh strategies for various types of queries.

In [32], the authors focus on adaptively adjusting the precision of cached approximations to achieve the best performance in dynamic distributed environments. The main goal is to minimize network traffic, therefore an approach is used whereby exact values are stored at the local nodes and appropriate precision interval approximations are stored at the cache nodes where the queries are posted, there by trading-off performance gains and decreased precision.

To handle applications that require event detection with low latency, [33] proposes an architecture, called PRESTO, that exploits a feedback-based, model-driven push approach to support queries in an energy-efficient, accurate and low-latency manner. PRESTO comprises a higher tier of sensor proxies and resources-constrained sensor nodes at the lower tier. A feedback-based model is constructed from correlations in the data acquired at each sensor. A sensor node at the lower-tier compares the sensed measurement against the model and pushes data to the proxy only when the difference between them exceeds a threshold. PRESTO assumes that each query specifies an error tolerance and a confidence interval. If the model prediction does not satisfy the query with the required precision, the proxy pulls actual measurements from the sensor nodes to process the query. The model parameters are periodically refined at the proxy, which cannot only result in an increase in the prediction accuracy but also reduce the number of pushes. The models currently used are temporal only.

In [34], the authors present a sensor database architecture called StonesDB that exploits developments in flash memory technology and emphasizes on storage-centric query processing in sensor networks. The architecture comprises two tiers, like PRESTO. The proxy caches summaries of sensor data (e.g., low resolution wavelet-based summaries of images) as well as the data recieved from the sensors, to enable more energy-efficient querying of sensors. The queries are recieved at the proxy, that determines how to handle queries with minimal energy cost. The goal of StonesDB is to resolve the trade-off between long-term storage and loss of query precision by designing strategies that should age out the least valuable data.

TSAR [35] is an in-network two-tier storage architecture. The sensor nodes are responsible for the data storage, metadata is stored at the proxies. Each sensor periodically sends a summary of its data to a proxy. Each TSAR adopts a novel adaptive summarization technique leading to finer-grained summaries that are only sent if many false positives are observed, thereby allowing the energy cost of metadata updates to be traded off the number of false positives. The TSAR index metadata in proxies uses an interval skip graph that efficiently supports spatio-temporal and value queries. The distributed index maintained at the proxies, helps in transmitting simple queries to explicitly identified storage locations in the lower tier. Moreover, TSAR does not have in-network tier that enables processing queries to take advantage of data locality.

DIFS [36] support efficient range queries on a single attribute while maintaining balanced load across nodes through the use of geographic hashing and spatial decomposition. DIFS constructs a geographical multi-rooted hierarchical range index that overcomes the problem of bottleneck as the root faced by most tree-based hierarchical approaches. Event information is stored at sensor nodes. In DIFS, event-to-sensor mapping is based on the K-D tree. DIFS's biggest contribution is that it scales well to large-scale networks. Two of the main problems from which DIFS suffers includes: (1) construction of K-D tree which might result in the formation of orphan regions (i.e., regions with no sensor nodes); (2) storage hot-spot problem. The formation of storage hot spots due to irregular sensor deployment or non-uniform event distribution is a key research problem that needs to be addressed. The storage hot-spot problem occurs when many events are mapped to relatively small number of nodes. Which may result in the following: (1) delaying queries due to the contention at the storage and the surrounding nodes, (2) hot spots nodes may become bottlenecks, due to quickly consuming energy, because of the high load of query and storing sensor readings. To overcome the node failure problem, it divides the events that are hashed to the same location among multiple mirror nodes. DIFS performance has not yet been evaluated on an experimental platform that collects and aggregates real data into high-level events. It does not handle range queries involving more than one attribute and dynamic repartitioning when a distribution changes over time.

KDDCS proposed in [38] use a distributed hashing index technique to solve range queries in a multi-dimensional space. KDDCS is proposed to solve the problems of storage hot-spots by avoiding formation of orphan regions and dynamically rebalancing the data region assigned to sensors. It presents a load-balanced storage scheme for sensor networks based on K-D tree, where nodes are assigned with bit-code identifiers that are related to their spatial location in the network. For constructing balanced K-D tree, a technique is presented which ensures that the number of sensor nodes on both parts of partition are equal during partitioning the geographical area. A K-D rebalancing algorithm is also presented to overcome the scenairios when events are not uniformally distributed.

Chunyu Ai et. al. presented an index-based historical data processing scheme in [39], for static WSN with the assumptions: (1) each node is location aware, and (2) any point in monitored area is covered by at least one node. Under this scheme, a network is divided into multiple grid with equal number of nodes. A hierarchial index tree is constructed by selecting leaders at multi-level. To give each node a chance to serve as index node, a tree switching period scheme has been introduced, allowing for maintaining load balancing among sensor nodes. The leader node of each cell is responsible for calculating the minimum, maximum and average values for each attribute in the cell. At each update interval, the leader node of each cell sends the computed values of that interval to its parent node in the index tree. Other work related to handling the storage hotspot problem includes [40]. A mathematical model is presented to find the optimal storage set of sensor nodes for the storage of events information. The base station is held responsible for computing the optimal storage set based on the prior information of the event and query distribution. For accuracy, such approach requires that the base station must be equipped with the up-to-date information about the type of events, queries and other required network metadata.

The authors presented a new storage scheme in [41]. This scheme stores the data in a node, close to the node where it is detected, and notifies the location information to the index node. The index node maintains an index list with all the nodes storing data for a given event type. An adaptive ring-based index scheme has been presented, which allows index nodes to form ring around the event region of particular type. The authors in [42] presented an adaptive storage switching policy which can dynamically choose between local storage, storage in the node closer to producer node, and external storage, depending on the the frequencies of user queries and events happening ratio. In this work, the queries are generated from several mobile sinks and the network is divided into multiple grids, and each grid decides locally which storage scheme to apply. Moreover, the full communication model as well as the policies to switch from one storage mode to the other are also presented.

4.2.5 Query Data using Precise Schemes

The fourth model can be referred to as in-network storage and focuses on storing the data close to the location where it is generated, i.e., inside the nodes themselves, in flash memory and the queries are disseminated into the network for processing.

In [43] geographic hash tables (GHT) are proposed as a data-centric storage [37] architecture; supporting aggregate and enumeration queries for event-driven applications. The approach allows data to be stored and accessed by name instead of sensor node ID. In a GHT, event names are hashed to geographic location where data associated with the events is stored. GHT uses geographical hierarchy to support load balancing. To avoid the problems of storage and communication that may arise due to storage hotspots, it divides the events that are hashed to the same location among multiple mirror nodes. Some issues that have not been investigated include varying node density and performance issues that may arise because of the non-uniformly distributed nodes.

The comb-needle approach proposed in [44] is another in-network storage architecture that uses an adaptive push-pull technique. In this approach, sink does not know the location of information in advance, no hashing technique is used. The goal is to minimize the number of communications required to get a complete response to a global discovery query such as ***”what locations have a temperature exceeding 90 degrees?”***. It allows each sensor node to push its data to a certain neighbourhood (transmitted vertically in both directions like a needle of a certain length) while the query request is disseminated to a subset of the network in the form of a comb with horizontal teeth. When the query is posed to the network by the sink, it traverses the network along such a comb. Once the event nodes are detected the information is transmitted to the sink along the shortest path. This strategy is recommended when the average number of queries, is less than or equal to the average number of events.

4.3 Storage Managers

Matchbox [16] is a simple sensor node filing system that is packaged with the TinyOS [45] distribution. It is a flat filing system that supports only sequential read and writes. Files are unstructured and represented internally as sequence of bytes. Checksum reliability mechanism is used to verify the integrity for each page of the file during recovery from a system crash.

Capsule [46] is fault-tolerant and provides more reliability by supporting check pointing and rollback of storage objects. Unlike, Matchbox, capsule filesystem, not only supports both read and write simultaneously to a file, but also operate on multiple files at the same time. Capsule filesystem provides useful additional features with a better overall energy profile and performance than Matchbox. It discusses data life efficiency but on some level of abstraction. *Efficient log structured flash file system* (ELF) [47], offers good reliability and wear leveling techniques for the flash file system. As compared to Matchbox, ELF offers more functionality, including random access, hierarchical directories and garbage.

Blackbook [48] is a flash file system. It is a software product of industrial research by Rincon Research Corporation. The architecture and design of Blackbook is not well documented. Files are broken into a linked list of nodes, each node containing some part of the file. Blackbook reassembles the nodes and present them as a continuous file. Blackbook also has a Dictionary file which is accessed a little differently than a binary file.

MicroHash [49], offers high performance indexing and searching capabilities by exploiting flash memory features such as asymmetric read/write and wear-out. MicroHash index is hash-based index structure designed for efficiently indexing temporal data for supporting value-based equality queries and time-based range and equality queries. The index structure has been designed for sensor nodes equipped with large flash memory, such as *RIverside Sensors* (RISE) equipped with several MBs of flash memory.

PIYA [50] storage manager designed for NAND-based flash memories allows to store the data sequentially in incremental fashion. For retrieving desired data, the table is constructed in the memory by reading the timestamps of every block. The procedure works by comparing the required timestamp with each page starting from the most currently written page. If the time stamp matches with any page then the required data items are checked. This procedure consumes long time and high energy, in case the flash memory is mostly/highly occupied. Varying *Aggressive data Quality Access Reference* (VAQAR) [51] is a storage management scheme, designed for motes with external large size NAND flash memory. In order to protect the critical data from deleting the storage manager allows to mark the data as undeletable by setting the byte in the first sector of each block containing critical data. The metadata regarding memory assigned to critical data, is stored in the form of map block. For efficiently retrieving desired data, the mapping table is constructed in memory from the metadata.

Matchbox maintains a pointer to each page of file on the flash memory, therefore, require a very large footprint in memory to keep track of these pointers. ELF uses linked list to represent open file in RAM; this list can be quite long if the file is long or has been updated repeatedly. Which will ultimately result in performance and reliability issues. In order to improve the write performance and increase the longevity of flash memory small write cache is maintained for a file in order to consolidate the writes to the same page [47][46][48][49][50][51]. Although buffering increases energy efficiency but decreases reliability and allows for opening only limited number of files because of limited RAM. **Table 2** highlights the support for the features (listed in Section 3.7) by the current storage managers.

Table 2. Features supported by Storage Managers

	MatchBox	**Capsule**	**ELF**	**BlackBook**	**VAQAR**	**Piya**	**Piyas**
Data Updation	No	NO	Yes	No	No	No	No
Device life efficiency	NO	Low	Low	Low	No	Low	Low
Performance and Reliability	Low	Medium	Low	Medium	Low	Low	Low
In-network data life efficiency	No	No	No	No	Low	Low	Low
Storage Efficiency	No	Low	No	No	Low	Low	Low
Searching Mechanism	Low	Medium	Low	Medium	low	low	low

5. Future Research Scopes and Open Issues

Sensor networks are by definition connected to the physical world and sensed data streams represent real world events. This implies the need for query processing to contend with near real-time issues. The stream is unbounded. This is made even more challenging by limited storage and the fact that storage management is expensive. To overcome the limited storage constraint, one approach is to apply a summarization technique founded on statistical methods or else to rely on compression methods to reduce storage needs. Thus, responses produced for queries that are posted on local stored data, are often approximate, instead of accurate, as they may be based on incomplete, or summarized, data. Which emphasizes that the spatio-temporal SNQP's, should resolve the trade-off between long-term storage and loss of query precision by designing strategies that should age out the least recently used or the least valuable data.

Since the nodes in a WSN and the communication edges formed by them map naturally to a finite set of points and a set of line segments, respectively, an algebraic approach to spatial analyses is possible. There exists work [13][14][15] that defines a discrete, and consistent geometric basis on which spatial data types (e.g., point, line, region) can be defined over WSNs. However, there are many environmental monitoring applications where the addition of a temporal dimension to yield an in-network spatio-temporal query processing over WSNs would be beneficial. For supporting in-network spatio-temporal historical queries, there exists related research on storage managers, but there is still a need to develop more sophisticated storage managers that can fullfil all the requirements for supporting historical spatio-temporal queries.

At the time of writing, no generally useful spatio-temporal query processing system has been proposed so far for sensor networks. Among the storage managers that are discussed in previous section, the implementations for Matchbox, Blackbook and Capsule are available. These implementations are functional on original motes (Mica, Mica2, Micaz) and currently there is no simulator that supports them. Programming WSNs require specialized knowledge, the scarcity of the resources puts a tight limit on code size, and debugging is cumbersome. Therefore, there is a need that current sensor network simulators provide support for simulating flash memory in order to test the effeciveness of the developed implementation over large scale network. There are a lot of open research issues in the field of spatio-temporal historical query processing for example, problems related to designing of sophisticated storage manager, support for large size flash memory by sensor nodes, spatial and temporal load-balancing schemes, spatio-temporal indexing, distributed spatial analysis, spatio-temporal and historical algebra, spatio-temporal SNQP, Spatio-temporal historical query language, distributed algorithms for spatial, temporal and historical operators, spatio-temporal SNQP, etc.

6. Conclusions

This paper highlighted some of the major challenges that are associated with the spatio-temporal and spatio-temporal historical query processing over wireless sensor networks. It also presented a survey of the research on different issues arising from a desire to support spatio-temporal and historical queries from various research communities have been surveyed. The problem of designing a general-purpose, distributed query processor handling spatio-temporal queries has not yet been addressed by the research community. Some work has been done to solve the challenges related to query processing over historical data, but there is still significant unexplored research space in this field.

References

[1] P. Bonnet, J. E. Gehrke, and P. Seshadri. “Querying the physical world” *IEEE Journal of Selected Areas in Communications*, vol.7, no.5, pp.10–15, Oct.2000. Article (CrossRef Link)

[2] Jane K. Hart and Kirk Martinez. “Environmental Sensor Networks: A revolution in the earth system science?” *Earth-Science Reviews*, no.78, pp.177–191, 2006. Article (CrossRef Link)

[3] Thomas Ulrich. “Wireless network monitors H_2O: System saves resources, increases yield in cabernet vineyard.” *Wines and Vines Magazine*, Jul.2008.

[4] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. “TinyDB: An acquisitional query processing system for sensor networks.” *ACM Trans. Database Syst*., vol.30, no.1, pp.122–173, 2005. Article (CrossRef Link)

[5] Yong Yao and Johannes Gehrke. “The cougar approach to in-network query processing in sensor networks.” *SIGMOD Record*, vol.31, no.3, pp.9–18, 2002. Article (CrossRef Link)

[6] Ixent Galpin, Christian Y. A. Brenninkmeijer, Alasdair J. G. Gray, Farhana Jabeen, Alvaro A. A. Fernandes, and Norman W. Paton. “SNEE: A query processor for wireless sensor networks.” *Distributed and Parallel Databases*, vol.29, no.1-2, pp.31–85, 2011. Article (CrossRef Link)

[7] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. “Wireless sensor networks for habitat monitoring.” in *Proc. of the 1st ACM international workshop on Wireless sensor networks and applications*, pp.88–97, 2002.

[8] Aline Baggio. “Wireless sensor networks in precision agriculture.” In *Workshop on Real-World Wireless Sensor Networks*, 2005.

[9] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. “Deploying a wireless sensor network on an active volcano. IEEE Internet Computing,” vol.10, no.2, pp.18–25, 2006.Article (CrossRef Link)

[10] A. Matese, S.F. Di Gennaro, A. Zaldei, L. Genesio, and F.P. Vaccari. “A wireless sensor network for precision viticulture: The NAV system. Computers and Electronics in Agriculture”, vol.69, no.1, pp.51 – 58, 2009. Article (CrossRef Link)

[11] F. J. Pierce and T. V. Elliott. “Regional and on-farm wireless sensor networks for agricultural systems in eastern washington” *Comput. Electron. Agric.*, vol.61, no.1, pp.32–43, 2008. Article (CrossRef Link)

[12] Wireless Module Portfolios. http://bullseye.xbow.com:81/Products/productdetails.aspx? sid=156, *2012*

[13] Farhana Jabeen. “Distributed Spatial Analysis in Wireless Sensor Networks”. *PhD thesis, School of Computer Science, University of Manchester*, 2011.

[14] Farhana Jabeen and Alvaro A. A. Fernandes. ”Distributed spatial analysis in wireless sensor networks.” *In ICPADS*, pp.558–567, 2010. Article (CrossRef Link)

[15] Farhana Jabeen and Alvaro A. A. Fernandes. ”Monitoring spatially referenced entities in wireless sensor networks.” In *Proc. of the 2010 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pp.398–403, 2010. Article (CrossRef Link)

[16] David Gay, Matchbox: A simple filing system for motes http://www.tinyos.net/tinyos-1.x/doc/matchbox.pdf, Aug.2003

[17] Markus Schneider. “Spatial data types for database systems: finite resolution geometry for geographic information systems,” *Lecture Notes in Computer Science*, vol.1288, 1997. Article (CrossRef Link)

[18] M. F. Worboys and M. Duckham. “Monitoring qualitative spatiotemporal change for geosensor networks”. *IJGIS*, vol.20, no.10, pp.1087–1108, 2006. Article (CrossRef Link)

[19] C. Farah, C. Zhong, M. F. Worboys, and S. Nittel. “Detecting topological change using a wireless sensor network.” *In GIScience*, pp.55–69, 2008. Article (CrossRef Link)

[20] J. Jiang and M. F. Worboys. “Detecting basic topological changes in sensor networks by local aggregation”. *In GIS*, pp.4, 2008. Article (CrossRef Link)

[21] Krishna Chintalapudi and Ramesh Govindan. “Localized edge detection in sensor fields.” *Ad Hoc Networks*, vol.1, no.2-3, pp.273–291, 2003.Article (CrossRef Link)

[22] Philip Levis and David Gay. “TinyOS Programming.” *Cambrige University Press*, 2009.

[23] Victor Shnayder, Mark Hempstead, Bor rong , Geoff Werner, and Matt Welsh. “Simulating the power consumption of large-scale sensor network applications.” *In SenSys*, pp.188–200, 2004. Article (CrossRef Link)

[24] .Adam Silberstein, Rebecca Braynard, and Jun Yang. “Constraint chaining: on energy-efficient continuous monitoring in sensor networks.” In *Proc. of the 2006 ACM SIGMOD*, pp.157-168. Article (CrossRef Link)

[25] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. “TAG: A tiny aggregation service for ad-hoc sensor networks.” *In OSDI*, 2002. Article (CrossRef Link)

[26] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. “Synopsis diffusion for robust aggregation in sensor networks”. In *SenSys*, pp.250–262, 2004. Article (CrossRef Link)

[27] Joseph M. Hellerstein, Wei Hong, Samuel Madden, and Kyle Stanek. “Beyond average: toward sophisticated sensing with queries.” In *Proc. of the 2nd international conference on Information processing in sensor networks*, pp.63-79, 2003. Article (CrossRef Link)

[28] Carlos Guestrin, Peter Bod´ Romain Thibaux, Mark A. Paskin, and, Samuel Madden. “Distributed regression: an efficient framework for modeling sensor network data”. In *IPSN*, pp.1–10, 2004.

[29] Adonis Skordylis, Niki Trigoni, Alexandre Guitton, and Birkbeck Coll. “A study of approximate data management techniques for sensor networks.” In *ICCSA*, pp.1–12, Jun.2006. Article (CrossRef Link)

[30] Deepak Ganesan, Deborah Estrin, and John Heidemann. “DIMENSIONS: Why do we need a new data handling architecture for sensor networks?” In Proc. of the ACM Workshop on hot topics in networks, pp.143–148, Oct.2002. Article (CrossRef Link)

[31] Minji Wu, Jianliang Xu, and Xueyan Tang. “Processing precision-constrained approximate queries in wireless sensor networks.” In *MDM*, pp.31, 2006. Article (CrossRef Link)

[32] Chris Olston, Boon Thau Loo, and Jennifer Widom. “Adaptive precision setting for cached approximate values.” In SIGMOD Conference, pp.355–366, 2001. Article (CrossRef Link)

[33] Ming Li, Deepak Ganesan, and Prashant Shenoy. “PRESTO: Feedback-driven data management in sensor networks.” In *Proc. of the Third Symposium on Networked Systems Design and Implementation*, May.2006. Article (CrossRef Link)

[34] Deepak Ganesan, Gaurav Mathur, and Prashant J. Shenoy. “Rethinking data management for storage-centric sensor networks.” In *CIDR*, pp.22–31, 2007.

[35] Peter Desnoyers, Deepak Ganesan, and Prashant J. Shenoy. “TSAR: a two tier sensor storage architecture using interval skip graphs.” In *SenSys*, pp.39–50, 2005. Article (CrossRef Link)

[36] Benjamin Greenstein, Sylvia Ratnasamy, Scott Shenker, and Ramesh Govindan. “DIFS: A distributed index for features in sensor networks.” *Ad Hoc Networks*, vol.1, no.2-3, pp.333–349, 2003. Article (CrossRef Link)

[37] Scott Shenker, Sylvia Ratnasamy, Brad Karp, Ramesh Govindan, and Deborah Estrin. “Data-centric storage in sensornets.” *Computer Communication Review*, vol.33, no.1, pp.137–142, 2003. Article (CrossRef Link)

[38] Aly, M.; Pruhs, K.; Chrysanthis, P.K. “KDDCS: A load-balanced in-network data-centric storage scheme for sensor networks.” In *Proc. of the 15th ACM international conference on Information and knowledge management*, pp.317–326, 2006. Article (CrossRef Link)

[39] Chunyu Ai, Ruiying Du, Minghong Zhang, and Yingshu Li. 2009. “In-Network Historical Data Storage and Query Processing Based on Distributed Indexing Techniques in Wireless Sensor Networks.” In *Proc. of the 4th International Conference on Wireless Algorithms, Systems, and Applications*, pp.264-273, 2009. Article (CrossRef Link)

[40] Xie, L.; L.C.; Chen, D.; Xie, L. “EEBASS: Energy-Efficient Balanced Storage Scheme for Sensor Networks.” In *GLOBECOM, IEEE,* pp.719–724, 2008. Article (CrossRef Link)

[41] Zhang, W.; Cao, G.; La Porta, T. “Data dissemination with ring-based index for wireless sensor networks.” In *Proc. of the 11th IEEE International Conference on Network Protocols, IEEE Computer Society*, pp.305, 2003. Article (CrossRef Link)

[42] Yu, G.J. Adaptive Storage Policy Switching for Wireless Sensor Networks. Wirel. Pers. Commun., vol.48, pp.327–346, 2009. Article (CrossRef Link)

[43] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. "GHT: A geographic hash table for data-centric storage." In *Proc. of the 1st ACM international workshop on Wireless sensor networks and applications*, pp.78-87, 2002. Article (CrossRef Link)

[44] Xin Liu, Qingfeng Huang, and Ying Zhang. Combs, Needles, "Haystacks: Balancing Push and Pull for Discovery in Large-Scale Sensor Networks." In *SenSys*, pp.122–133, 2004. Article (CrossRef Link)

[45] Philip Levis, Samuel Madden, David Gay, Joseph Polastre, Robert Szewczyk, Alec Woo, Eric A. Brewer, and David E. Culler. "The emergence of networking abstractions and techniques in TinyO*S." In NSDI*, pp.1–14, 2004. Article (CrossRef Link)

[46] Gaurav Mathur, Peter Desnoyers, Deepak Ganesan, and Prashant Shenoy. "Capsule: an energy-optimized object storage system for memory-constrained sensor devices". In *SenSys*, in *Proc. of the 4th international conference on Embedded networked sensor systems*, pp.195–208, 2006. Article (CrossRef Link)

[47] Hui Dai, Michael Neufeld, and Richard Han. "ELF: an efficient log-structured flash file system for micro sensor nodes." *In Proc. of the 2nd international conference on Embedded networked sensor systems*, pp.176–187, 2004. Article (CrossRef Link)

[48] Rincon Research Corporation, BlackBook, tinyos-1.x/contrib/rincon/tools/java/com/rincon/blackbook

[49] Demetrios Zeinalipour-Yazti, Song Lin, Vana Kalogeraki, Dimitrios Gunopulos, and Walid A. Najjar. Microhash "An efficient index structure for flash-based sensor devices." In *Proc. of the 4th conference on USENIX Conference on File and Storage Technologies*, pp.3–3, 2005.

[50] Sanam Shahla Rizvi and Tae-Sun Chung. 2008. "PIYA - Proceeding to Intelligent Service Oriented Memory Allocation for Flash Based Sensor Devices in Wireless Sensor Networks." In *Proc. of the 2008 Third International Conference on Convergence and Hybrid Information Technology,* vol.01, pp.625-630,2008. Article (CrossRef Link)

[51] S.S. Rizvi and T.S.Chung. "VAQAR: Flash memory based long term in-network vital data sustainability and availability for data centric wireless sensor network applications." In *Proc. IEEE Youth Conf. Inf., Computing and Telecom*., pp.363–366, 2009, Article (CrossRef Link).

Farhana Jabeen obtained her PhD from Manchester University in 2011. Currently, she is working as Assistant Professor at COMSATS Institute of Information Technology (CIIT) Islamabad, Pakistan. She has previously worked in the area of distributed query processing, being a member of the team that designed and developed the SNEE (Sensor NEtwork Engine) query optimizer for sensor network.

She has recently contributed to distributed spatial analysis in wireless sensor networks. She has previously held academic post in Fatima Jinnah Women University, Pakistan, where she got various opportunities to act in various other roles including head of department and researcher. As a researcher she worked on several projects. Her other research interests include query processing in wireless sensor network, and distributed computing.

Sarfraz Nawaz received a bachelors in Electrical Engineering from University of Engineering and Technology Lahore in 2003 and a PhD from University of New South Wales Australia in 2008. Since then he has been actively involved in wireless sensor networks research. After his PhD, he joined the Computer Laboratory at University of Oxford where he developed robust localization algorithms for cluttered and multi-path environments. He then joined the Computer Laboratory at University of Cambridge where he is working with wireless sensor networks for infrastructure monitoring as part of the Centre for Smart Infrastructure and Construction. His other research interests include mobile phone sensing and ubiquitous computing.

Sadaf Tanvir is serving as Assistant Professor at COMSATS Institute of Information Technology (CIIT) Islamabad, Pakistan. She had her BS from the same institute and her MS from University Paul Sabatier, Toulouse, France. She has availed Pakistan’s Higher Education Commission (HEC) scholarship for MS leading to PhD. She holds PhD degree from University of Grenoble, France, 2010. Her research interests include wireless sensor networks, routing and MAC in wireless sensor networks and the energy optimization in these domains.

Majid Iqbal Khan received his undergraduate degree with majors in Mathematics and Physics (2001) and Master degrees in Software Engineering (2004). He obtained a PhD in Wireless Sensor Networks from University of Vienna, Austria in 2009. He joined COMSATS Institute of Information Technology in 2009, where he is currently an Assistant Professor. His research interest includes wireless sensor networks, MANETS and internet of things.