Information Fusion 15 (2014) 32–43

Contents lists available at SciVerse ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Spatial query processing in wireless sensor networks – A survey

Rone Ilídio da Silva [a,b,*], Daniel Fernandes Macedo [b], José Marcos S. Nogueira [b]

[a] Universidade Federal de São João del-Rei, Campus Alto Paraopeba, Ouro Branco, MG, Brazil
[b] Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil

ARTICLE INFO

Article history:
Received 25 April 2010
Received in revised form 8 May 2012
Accepted 23 August 2012
Available online 18 September 2012

Keywords:
Spatial query
Wireless sensor network
KNN
Window query

ABSTRACT

Wireless sensor networks (WSN) are particularly useful for obtaining data concerning events limited to a well-defined geographic region, such as a disaster site or a malfunctioning subsection of a factory plant. Such applications typically use spatial queries, which are SQL-like queries where location constraints are imposed on the collected data. Further, spatial queries allow changing the set of nodes (the region of interest) at runtime. This work surveys spatial queries in WSN. Due to the particular energy and resource constraints of WSN, spatial queries are performed by mechanisms having several stages, each of them implemented using localized distributed algorithms. This article categorizes the existing strategies for each stage, in order to ease the understanding of the state of the art. Finally, we analyze the most recent works on spatial query processing, identifying which classes of algorithms are used on each stage.

1. Introduction

Wireless sensor networks (WSN) are formed by many small devices that sense, process, store and transmit environmental data. These devices have limited power supply, and usually it is not feasible to replace their batteries because WSN may be deployed to monitor inhospitable environments [1]. As the technology matured, WSN became more than periodic data gathering tools. Now, they are data gathering tools able to answer users' queries [2]. Several works model WSN as distributed databases and define energy aware mechanisms for in-network query processing, such as TinyDB [3]. They process declarative SQL-Like queries, which simplify the way that the users obtain data from the WSN.

Recently, several researchers have worked on node location for WSN [4]. In-network query processing mechanisms can use location data to answer a special type of query called *spatial query*. In these queries the users' interests are expressed by geographical predicates, such as "the temperature collected by nodes in a region" or "the humidity collected by nodes closest to a point".

Spatial queries are database queries supported by geodatabases [5]. These queries differ from traditional queries in two main points. First, they incorporate geometry data types, such as points and polygons. Second, these queries consider the spatial relationship between the defined geometries, such as a point *inside* a polygon or a polygon that *overlaps* another. WSN can be modeled as distributed geodatabases in order to process queries containing spatial information. Those queries would request data collected inside a region (called *region of interest*) or nearest to a point defined by the user (called *query point*).

Thus, spatial queries in WSN usually rely on distributed and localized algorithms to reduce their resource usage and energy consumption: queries are forwarded towards the region of interest based on the neighborhood information of a few hops; the selection of the nodes to be activated for sensing is performed by the nodes inside the region of interest; finally, the sensed data is aggregated during the return path from the region of interest to the sink, hop by hop. Thus, information fusion is one of the steps performed in spatial query processing.

Due to the energy constraints of the sensor nodes, the collected data are processed during the forwarding to the user, using information fusion techniques to summarize them and to eliminate redundancy, in a process known as *in-network query processing* [6]. In-network processing techniques potentially reduce the energy consumption and traffic load on the network, however they place a heavy burden on the spatial query mechanisms, because they increase the penalties of packet losses and late deliveries. Thus, spatial queries must attempt to reduce the amount of packet losses, even though the network is prone to failures due to bad wireless links, node failures, energy depletion, among others.

This work presents a survey of spatial query processing in wireless sensor networks. We divide spatial query processing in six stages and describe the existing solutions for each stage. This division simplifies the understanding of spatial queries and facilitates the proposal of new solutions. We also survey the literature, categorizing the algorithms for each stage according to their most

* Corresponding author at: Campus Alto Paraopeba, Universidade Federal de são João del-Rei, Rodovia MG 443, KM 7, Ouro Branco, MG 36400-000, Brazil. Tel.: +55 31 37413280.

E-mail addresses: rone@ufsj.edu.br (R.I. da Silva), damacedo@dcc.ufmg.br (D.F. Macedo), jmarcos@dcc.ufmg.br (José Marcos S. Nogueira).

1566-2535/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.inffus.2012.08.006

important properties. Further, we present how each spatial query processing mechanism in the literature implements these stages.

This article is organized into five sections. Section 2 describes the six stages for spatial query processing. Section 3 presents how the spatial query processing mechanisms in the literature perform each stage. Section 4 presents the main contributions and drawbacks of spatial query processing mechanisms found in the literature. Section 7 discusses the importance of information fusion on spatial query processing. Section 6 presents the open research challenges in spatial query processing. Finally, Section 7 shows our conclusions.

2. Spatial query processing stages

In order to better study spatial queries, we will analyze them in stages. The related works define different numbers of stages for spatial query processing in WSN [7–10]. However, these stages can be subdivided into simpler stages, as proposed in our previous work [11]. We analyzed spatial queries in six stages: Pre-Processing, Forwarding, Dissemination, Aggregation, Sensing and Return. Fig. 1 illustrates them.

2.1. Pre-Processing stage

In the Pre-Processing stage, queries are formatted in order to be transmitted by the sensor nodes. It is performed in the user's computer, since usually this computer has more resources than the sensor nodes. Normally, queries are written in a declarative language based on SQL [3]. Thus, queries must be transformed into a sequence of bytes to be transmitted to the nodes. Moreover, packets transmitted by traditional WSN using 802.15.4 usually have at most 88 bytes [12]. As a consequence, this sequence must be as small as possible, since a big sequence may need several packets to be transmitted and, consequently, would consume more energy during its processing.

Before the Pre-Processing stage, the user must define the parameters of the spatial query. For example, it selects a region of interest, which collection method it will use (a snapshot of the environment or a continuous sampling) and its parameters (e.g. the periodicity among collections in continuous sampling). Those parameters will depend on the application and its requirements. Hence, the Pre-Processing stage must performs application-independent tasks, such as representing the information in a most suitable manner, in order to make the query more efficient and so it takes up less packets.

2.2. Forwarding stage

In this stage, queries are forwarded from the *Originator* (the first node to receive the query in the network) to the region of interest. This is the main difference between conventional query processing and spatial query processing in WSN. In conventional query processing, queries are disseminated to all nodes in the WSN by Flooding [13]. In spatial query processing, queries are first forwarded and then disseminated only to the nodes within the region of interest. Hence, the task of the Forwarding stage can be performed by adapting one of the routing protocols found in the literature [14].

Another important characteristic to be considered in Forwarding is node location. Spatial query processing mechanisms assume that nodes know their location. Location-based routing protocols use this information to create their routes, consuming less energy than the protocols that do not employ location information [14]. Thus, these protocols are usually the best choice for the Forwarding stage. Further, each protocol is adapted to a specific network mobility pattern. Hence, the spatial queries processing mechanisms need to choose the routing protocol that better adapts to the target network.

2.3. Dissemination stage

In the Dissemination stage, the query is disseminated to all nodes within the region of interest. Dissemination algorithms try to minimize the number of packets transmitted by the WSN in order to reduce the energy consumption. The Dissemination strategy defines the type of the spatial query. We found in the literature two types of spatial queries, called window query and k-nearest Neighbor query (KNN).

Window queries are the most common type of spatial queries. The user defines a region of interest (called *window*) and asks for data collected by the sensor nodes inside this region. Windows are defined by rectangles [15,8,16], circles [9] or polygons with irregular contours [11,17]. These queries are also called range queries[1] [8]. Fig. 2a illustrates a Window query in a WSN.

K-Nearest Neighbor queries (KNN) retrieve data from the K nodes closest to a point defined by the user, called *query point* [18–20]. Fig. 2b illustrates a KNN query. We also found in the literature a type of query called *Nearest Neighbor queries (NN)*. It is a special case of KNN, with $K = 1$ (such as [2]). Hence, we focus on KNN queries.

2.4. Sensing stage

In the Sensing stage, the nodes within the region of interest collect the data required by the query. The operation of this stage is heavily dependent on the application, and as such are not part of spatial query mechanisms. We identified three high level the tasks that are usually performed in this stage. These tasks are *raw data collection*, *local information fusion*, and *compression*.

The first task entails the collection of the data itself. This may involve the use of physical or virtual sensors [21]. In the field of context-aware systems, physical sensors are hardware that collect readings related to the environment, such as humidity and temperature, while virtual sensors may collect data referring to the current node's state, such as its remaining energy and processing capabilities. Physical readings are useful for monitoring and actuation, while virtual readings are often employed for management tasks, in order to configure the network according to its current operational state [22].

After the data has been collected from one or more sources, nodes may perform local information fusion. This is the first stage in spatial queries where information fusion may take place. The node may use the raw data to make inferences based on the readings of multiple sensors. Further, in applications that require a coarse spatial granularity of the data [23], the nodes may employ strategies to reduce the number of nodes that sense the environment without compromising the quality of the reported data [24].

Finally, the node may employ data compression strategies to reduce the size of the information, once the irrelevant data has been suppressed in the previous tasks. This tasks deals with information coding, rather than the content of the information to be transmitted. Techniques such as delta coding, Huffman's method and others could be employed [25].

2.5. Aggregation stage

In the Aggregation stage, the collected data are transmitted from the nodes in the region of interest to the *Aggregator*, the node that calculates the query result. In this stage, nodes are able to perform information fusion in the collected data. There are two possible information fusion strategies: naive and in-network processing. In a naive strategy, all the nodes within the region of interest

[1] In traditional database literature, range queries select data between an upper and lower boundary, however these boundaries may not be related to spatial constraints.

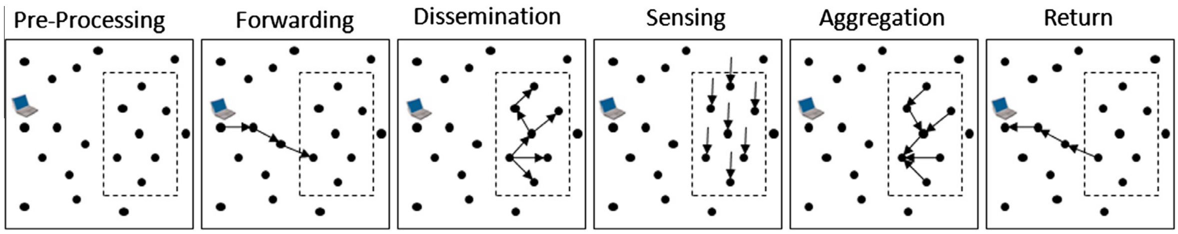

Fig. 1. Six states of spatial query processing.

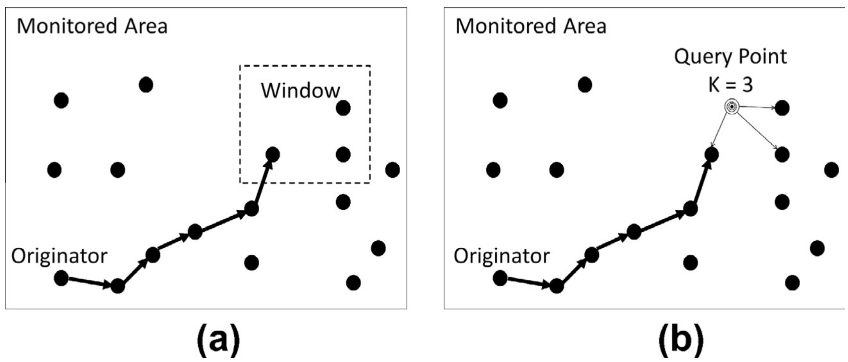

Fig. 2. (a) Window query. (b) K-nearest neighbor query.

would send their data to the Aggregator using multi-hop routes. Furthermore, nodes next to the Aggregator consume more energy than the others, because all the data needs to pass through them. Due to the significant energy constraints of sensor networks, it is advisable to reduce the amount of data sent during the Aggregation stage. Although we usually obtain better summarization and inferences by having the entire set of observations, it is commonplace to trade-off precision with energy consumption during the Aggregation stage.

A possible strategy is to combine the sensing data during the propagation (also known as *in-network aggregation* [6]) in order to reduce the number of packets transmitted. In this approach, the information is refined in steps, where each node attempts to process the information received so far, which in turn usually reduces the size of the data to be transmitted.

2.6. Return stage

In the Return stage, the Aggregator sends the query result back to the Originator. Most mechanisms use either the same route created during the Forwarding stage or the same protocol used in this stage. However, in mobile WSN the route used in the Forwarding stage may not be available to the Return stage since the nodes can move during the dissemination. Further, the same protocol may not be used because the place where the query was issued probably is not the place where the query must be delivered, since the Originator is mobile.

Among the analyzed works on spatial query processing, queries also can be classified by the result type. We found three types: snapshot, spatiotemporal and longlife. *Snapshot* queries retrieve data just once. Sensor nodes sense the environment, transmit the collected data and wait for a new query. *Spatiotemporal* queries specify a window of time (a lower and an upper limit) for the data to be retrieved. These queries manipulate data stored at the flash memory of the sensor nodes or data collected periodically by the nodes, stored at the sink. *Longlife* queries generate periodic answers. In these queries, the users specify the reporting period and the query lifetime.

3. State of the art on spatial query processing in WSN

In this section we present a set of solutions found in the literature for each processing stage. Clearly, we can find other solutions that could be used in one of the stages. However, these solutions are not analyzed here, since they are not adopted by the spatial query processing mechanisms found in the literature. An example is the Forwarding stage. Several protocols can be used in this stage, such as [26,27]. However, we only analyze those that are used by existing spatial queries processing mechanisms.

At the end of this section we also present an overview of distributed spatial indexes created to process spatial queries. In these mechanisms, the Forwarding and Dissemination stages and the Aggregation and Return stages are performed by the same algorithm. These techniques are analyzed separately.

3.1. Pre-Processing stage

The Pre-Processing stage was proposed in our previous work [11]. In the literature, works on spatial query processing define regions for data collection using rectangular or circular shapes, which need only two points to be represented in the queries. None of these works refer to queries where the region of interest has irregular contours. This kind of representation is important because it can represent real objects plotted in maps or satellite photos.

We considered spatial queries that have regions of interest in the form of polygons with irregular shapes. Our Pre-Processing algorithm reduces the number of points needed to represent the region of interest in the queries. Hence, it also reduces the number of network packets to transmit the query between two neighbor nodes. Consequently, WSN transmit less packets and consume less energy. Fig. 3a illustrates a polygon that represents a lake plotted on a satellite photo. Fig. 3b shows the polygon with a reduced number of points.

The polygon simplification decreases the quality of the query result, because some sensor nodes inside the original region of interest do not participate in the query processing. Moreover, some

Fig. 3. (a) Polygon representing a lake. (b) Simplified polygon representation.

nodes outside this region participate. Hence, there is a tradeoff between energy saving and query accuracy [11].

3.2. Forwarding stage

Several routing protocols could be adapted to perform the Forwarding stage, some of them using the nodes' location to create routes. This kind of protocol is named location-based routing protocols or geographic routing protocols [28]. The spatial query processing mechanisms found in the literature are based on the following location-based protocols: Greedy, GPSR and SQR. However, [29,19] presented spatial query processing mechanisms that use Flooding to forward the query to the region of interest. In these mechanisms, all the nodes receive the query, but just those within the region of interest process it. This subsection analyzes the advantages and disadvantages of Flooding to forward spatial queries. Then, we present an overview of the location-based routing protocols used in spatial query processing.

3.2.1. Flooding in WSN

Flooding is a strategy to disseminate data to all nodes in a network. It is used for spatial query processing in [29,19] and compared to the algorithms proposed in [30,7]. Its advantages are: if there is a path between the Originator and the region of interest, the query will always reach it; it does not demand many computational resources; the query packet converges quickly to the region of interest; the Forward and Dissemination stages are performed together, and; it builds a routing tree that can be used to return the query result.

The disadvantages are: it consumes too much energy, because nodes can receive the same query more than once (from all its neighbors) and the nodes outside the region of interest receive and retransmit the query unnecessarily, and; the query reaches the region of interest through more than one node, thus there is more than one route to send back the query result, impairing the aggregation process; an efficient aggregation process is performed only inside the region of interest [8].

In [7], the authors compared two strategies to process spatial queries. In the first, they used Flooding to forward the queries. In the second, they used the Greedy protocol (Section 3.2.2) for Forwarding and Restricted Flooding for Dissemination (Section 3.3.1). The experiments showed that Flooding consumes more energy than the second strategy in all the analyzed scenarios.

3.2.2. Greedy protocol

The Greedy Protocol is the most simple and intuitive location-based protocol. Nodes forward the packet to the neighbor closest to the destination. This strategy is very simple and requires mild computational resources. However, this protocol depends on a neighbor table, which contains the location of all neighbors of a node. This table needs to be updated periodically, hence it consumes part of the network energy supply, mainly in WSN with mobile nodes where topologies changes are frequent. However, in static WSN it presents reasonable energy consumption. The spatial query processing mechanisms described in [11,7] use the Greedy protocol in the Forwarding stage.

The Greedy protocol, however, does not guarantee packet delivery. If a node has no neighbor closer to the region of interest than itself, the query will be dropped. This problem is called as *hole in the network* [31] and is illustrated in Fig. 4a. The query reached a node that is the closest node to the region of interest and it does not have any neighbor closer than itself or inside this region. Hence, the query will be dropped.

3.2.3. GPSR – Greedy Perimeter Stateless Routing

GPSR is one of the most referenced location-based routing protocols [32]. It forwards the packets like the Greedy protocol. However, packets are not dropped if they reach a hole in the network. GPSR constructs a planar graph with neighborhood information [31]. Packets near a hole are transmitted over the contour of the faces of the planar graph. Fig. 4b illustrates a query contouring a face to reach the region of interest. Hence, this protocol delivers more packets than the Greedy protocol and has the same energy consumption. However, it also depends on a neighbor table. The spatial query processing mechanisms described in [18,2,20,30] use GPSR to forward the queries.

3.2.4. SQR – Spatial Query Routing

Xu and Lee [10] proposed SQR, a routing protocol for WSN with mobile nodes and having nodes that can transit to sleep mode. This protocol performs two tasks to transmit the query from a node A to the next hop in the route. First, A searches for a volunteer among its neighbors. Second, A sends the query to this volunteer. A searches for a volunteer broadcasting a message with query information. When a candidate node receives this message, it holds the query for a time period calculated from its distance to the anchor point (the center of the window) and its energy level. The node B (the neighbor of A closest to the anchor point and with a good energy level) is the first node to answer. So, B sends an ACK_Forwarder message. When A receives this message, A chooses B as the next hop and sends the query to B. The other neighbors of A also receive the ACK_Forwarder message sent by B and realize that they are not the chosen volunteers. Moreover, the neighbors of B also receive the message. Hence, they operate like B to choose the next node that will forward the query in the route. This operation is repeated until the query reaches the region of interest. The authors of [10] described SQR, but did not evaluate its performance.

3.2.5. Analyzing the location-based routing protocols

The analyzed works on spatial query processing showed that Flooding can be used by static or mobile networks, but it has higher

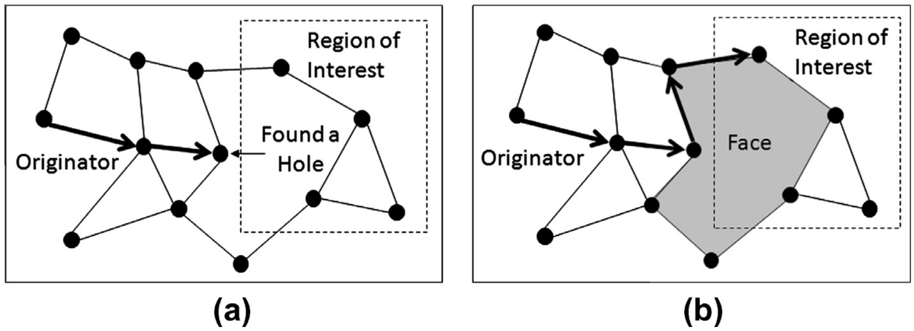

Fig. 4. (a) Hole in the network. (b) The query bypasses the hole.

energy consumption. SQR is a good option to the Forwarding stage, but it was not evaluated. Hence, we conclude that these protocols are not the best choice for the Forwarding stage. Greedy and GPSR protocols are better options, since they do not require much computational resources and consume little energy. Moreover, GPSR has more advantages since this protocol can avoid holes in WSN. However, both depend on the neighbor table, whose maintenance may consume too much energy.

3.3. Dissemination stage

In the Dissemination stage, the query is transmitted to all nodes within the region of interest. In the literature of spatial query processing, we found four protocols: Restricted Flooding, itinerary-based, SPA and WinDeph. It is important to mention that the Aggregation stage depends on the manner that dissemination was performed. In some cases, these stages are performed by the same algorithm, such as in itinerary-based protocols [30].

3.3.1. Restricted Flooding

Restricted Flooding is the most referenced protocol to disseminate spatial queries. This protocol is used in the mechanisms described in [19,11,17,7]. When a node *A* receives a query, it verifies if it is inside the region of interest. If true, it will broadcast the query to its neighbors, which saves *A* as its parent node. Nodes outside the region of interest will drop the query. This protocol creates a routing tree inside the region of interest with the Coordinator as the root. This tree is used in the Aggregation stage to calculate the query result.

A drawback of this protocol is the energy consumption. The network consumes a substantial part of its energy with unnecessary query transmissions and receptions. It is because the nodes close to the region of interest also receive the query from the nodes inside this region. Moreover, each node receives the query from all its neighbors during the dissemination. In our previous work [17] we proposed a dissemination strategy that turns off the radio of the nodes in predefined periods of time during the Dissemination stage. This strategy avoids that nodes participating in the Restricted Flooding receive a query from more than one source in order to decrease the energy consumption.

3.3.2. WinDepth

WinDepth was proposed in [7]. It disseminates the query node by node until all have received the query. When a node receives a query to disseminate, it adds its node identifier in the query header in order to create a return path for the query result. Then, it selects a neighbor located inside the region of interest that has not received the query yet and sends it to this neighbor. When this neighbor returns the partial query result, the node verifies again if other neighbors are also inside the region of interest and have not yet received the query. If such neighbor exists, the node sends the query to this neighbor and waits for its answer. This process is repeated until all the node's neighbors located inside the region of interest have answered the query. When all neighbors send their data, the node merges this information with the information collected by itself and sends the partial query result to the neighbor that sent the query to it.

3.3.3. Itinerary-based protocols

An itinerary is a sequence of nodes within the region of interest that forms a path over which the queries are forwarded. Each node that receives a query follows these steps: asks its neighbors to sense environmental data, senses its own data, receives the information from its neighbor and retransmits the query and the partial query result to the next node in the itinerary. When the query reaches the last node in the itinerary, the result is forwarded to the Originator.

Xu et al. [30] proposed a mechanism to process spatial window queries called IWQE (Itinerary-base Window Query Execution). Fig. 5 shows an example of an itinerary created by this mechanism. During the query dissemination on the itinerary, IWQE tries to avoid holes in the network as follows. If the query reaches a hole, the query propagation switches to a perimeter forwarding mode on the progressively closer face of the planar graph, like in GPSR. However, IWQE does not guarantee that the query always reaches the end of the itinerary. If the query in the face of the graph reaches a node outside the region of interest, the process is stopped and the partial query result is sent to the Originator.

Dissemination by itinerary is most used in KNN query processing because the itinerary facilitates counting the number of nodes that processed the query. We found three mechanisms that process KNN queries and use itineraries: IKNN [19], DIKNN [18] and PCIKNN [20]. The mechanisms based on itinerary differ from each other mainly in the format of the itineraries. DIKNN divides the region of interest into cone-shapes areas centralized at the query point. In each area, an itinerary is created, as shown in Fig. 6a. PCIKNN also divides the region in cone-shapes and creates several

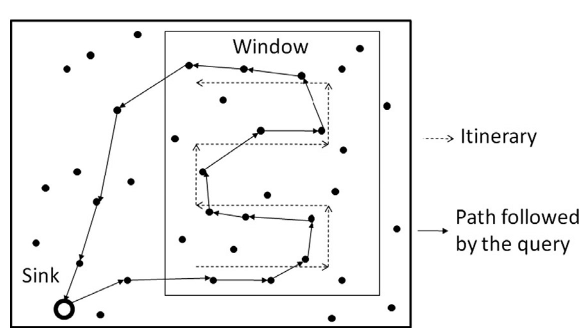

Fig. 5. Window query processing based on itinerary.

itineraries. However, the itineraries have a different format, as shown in Fig. 6b.

IKNN proposed two formats for the itinerary. The first is called Spiral Itinerary. The query dissemination starts in the node closest to the query point, following an itinerary in the form of a spiral, as shown in Fig. 6c. Queries have a field that stores the number of nodes that answered the query. When this number reaches K, the query dissemination is stopped and the result is sent back to the Originator. The second format is called Parallel Itinerary. The dissemination starts in the same node, but queries follow two itineraries, shown in Fig. 6d. The query also stores the number of nodes that answered the query. Neighbor nodes in different itineraries sum their counts to calculate the total number of nodes that answered the query. The parts of the itineraries where these nodes send their count to each other are shown in Fig. 6d.

In those works, the region of interest is circular, because it facilitates to identify the nodes closest to the query point. The range of the region of interest is calculated based on the WSN density. During the Forwarding stage, each node puts in the query packet its number of neighbors. The Coordinator uses this information to define the region of interest that probably contains K nodes.

3.3.4. SPA – Spatial Propagation and Aggregation

SPA was proposed in [10] to disseminate queries in mobile WSN. It first divides the region of interest in sub-regions and then disseminates the query. Fig. 7 illustrates this division. The region of interest is divided in four sub-areas (w1, w2, w3 and w4) to form the level 1. Each sub-area is also divided in four sub-areas to form the level 2 (the sub-area w2 is divided in w21, w22, w23 and w24). The number of levels is defined based on the node density. Each level has a leader, which is the node closest to the center of the sub-area. This structure creates a tree, in which the root is the leader of the level 1, the other leaders are the intermediary nodes and the leafs are the remaining nodes. This type of tree is called Quad-Tree [33].

A query is first disseminated to the leader in level 1. This node disseminates it to the leaders in the level 2. This process is repeated until the query reaches the leaders in the last level. These leaders disseminate the query to all nodes in their sub-region. However, SPA considers mobile nodes and nodes that turn off their radios to save energy. So, if a leader moves out of its area or switches to sleep mode, the other nodes negotiate to find a new leader. Hence, nodes need to verify periodically in which sub-area they are and who are their leaders.

3.3.5. Analyzing the Dissemination protocols

In Window and KNN query processing itinerary-based techniques present the best results in static and mobile WSN. In [30], a window query mechanism that uses an itinerary-based protocol to disseminate the query was compared with three other mechanisms: Restricted Flooding, MBR (Section 3.6.1) and Flooding. In most of the experiments, the mechanism with itinerary-based dissemination presented the smallest energy consumption and query latency. Moreover, the query result accuracy was similar to the analyzed mechanisms. In [19] the authors compared three KNN query processing mechanisms. They used itinerary-based, Restricted Flooding and Flooding to disseminate the query. The itinerary-based mechanisms also presented the smallest energy consumption, the smallest query latency and similar query accuracy.

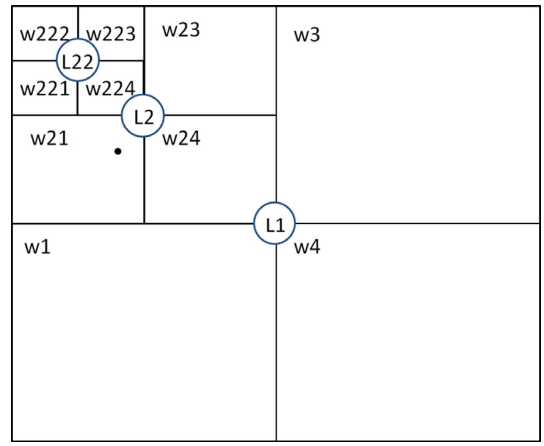

Fig. 7. The window divided in sub-areas.

3.4. Aggregation stage

The works on spatial query processing consider five aggregation operators: *avg*, *count*, *sum*, *max* and *min*, however other methods could also be employed in this stage [34]. As mentioned above, one may employ a naive strategy, where the Aggregation functions are applied to the data only at the Coordinator, or in-network aggregation strategies, where local aggregation occurs at each node. The latter is more common in WSN since it consumes less energy, even though it may lead to less accurate information.

The choice of the aggregation strategy depends on the algorithm used to disseminate the query. When Restricted Flooding is used, a routing tree can be used to aggregate the data. The leaf nodes start the aggregation process by sending their data to their parent. The parent node receives these data, merges it with its own data, applies the aggregation operator and sends the result to its parent. This process is repeated until the Coordinator receives the data from all its children, calculates the query result and sends it to the Originator.

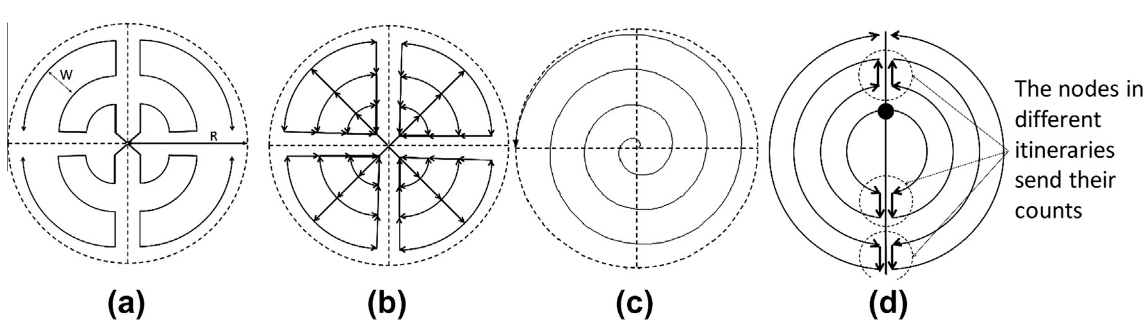

Fig. 6. (a) Itineraries created by DIKNN. (b) Itineraries created by PCIKNN. (c) Spiral itineraries created by IKNN mechanism. (d) Parallel itineraries created by IKNN mechanism.

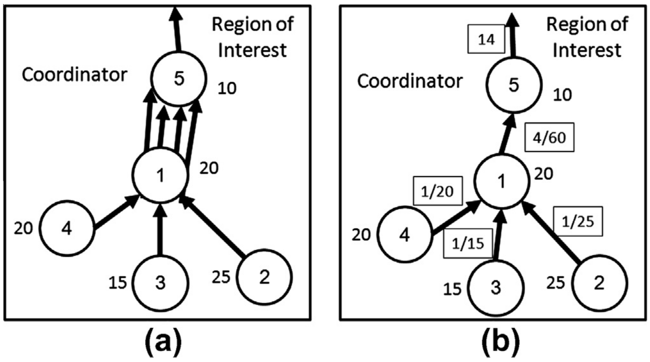

Fig. 8. (a) Naive strategy to aggregate data. (b) In-network Aggregation using routing tree.

Fig. 8 shows two examples of aggregation using the *avg* operator. Other aggregation operators are performed in the same manner. The values next to the nodes represent the sensed data that was sent to parent node. Arrows represent packet transmissions. Fig. 8a shows the naive strategy, while Fig. 8b shows an example using in-network aggregation. We can see in (a) that nodes inside the region of interest sent eight packets to perform the aggregation. In (b), nodes sent only five packets. This type of aggregation is used in the spatial query processing mechanisms described in [11,17,7,19].

One of the main concerns of the aggregation stage is to reduce the amount of packet losses. This is specially important when nodes employ information fusion strategies, since the loss of a packet is more critical for the accuracy of the application [35]. Further, the aggregation strategies attempt to mitigate packet losses. There are two main causes of packet losses in the aggregation stage, which are packet collisions caused by concurrent transmissions, and forwarding failures caused by node mobility. Besides the amount of packet losses, the performance of the aggregation stage is also measured by its energy consumption and its execution time. The former is important due to the resource constraints of the nodes, while the latter is more critical for time-sensitive applications.

When Flooding is used to disseminate, the aggregation process can transmit more packets than necessary. In this case, part of the aggregation is performed outside the region of interest, as shown in Fig. 9a. Node 5 finished the aggregation process and the WSN transmitted nine packets. In Fig. 9b the aggregation process is performed only inside the region of interest. In this example, the WSN transmitted only seven packets. The mechanism described in [29] disseminates the queries by Flooding and aggregates as in Fig. 9a.

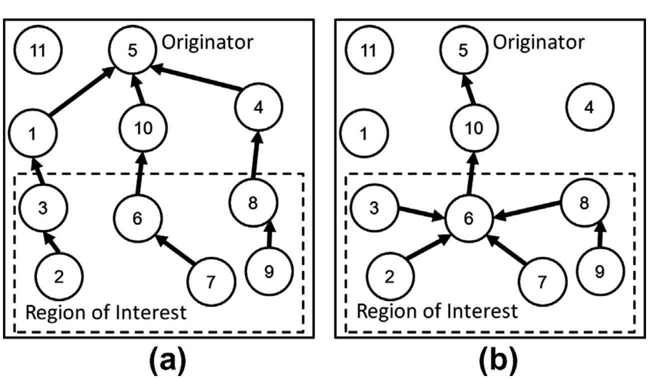

Fig. 9. (a) Aggregation performed outside the region of interest. (b) Aggregation only inside the region of interest.

Moreover, Flooding-based strategies tend to produce more collisions, which may lead to loss of information.

The mechanisms that disseminate the query by itineraries perform the Dissemination and the aggregation together. Nodes that receive the query on the itinerary ask their neighbors to sense and to transmit the sensed data. Then, it receives this data, merges it with its own data, applies the aggregation operator and forwards the partial result and the query to the next node on the itinerary. Itinerary-based aggregation tends to produce less collisions that flooding-based aggregation, but this comes at a higher response time. The mechanisms described in [20,30,18,19][2] use itineraries.

3.4.1. Analyzing Aggregation strategies

In-network aggregation strategies substantially reduce the energy consumption during the Aggregation stage. However, all the aggregation strategies consume more energy in mobile WSN, as shown in [30,19]. The trees created by Flooding or Restricted Flooding need to be maintained in order to guarantee the delivery of the partial query results to the Coordinator. Hence, in mobile WSN the nodes continuously verify if the communication with their parents is possible. If not, they look for another parent. When the aggregation occurs jointly with the dissemination (as in protocols based on itineraries) node mobility also requires periodic link verification because nodes need to establish the next node on the itinerary. This problem is aggravated in longlife queries, because the queries run in the WSN for a larger period of time.

3.5. Return stage

In the Return stage, the Aggregator sends the query result to the Originator. The majority of the works on spatial query processing use GPSR to perform this stage, such as [19,30,20,2,18]. Other mechanisms use the opposite route used in the Forwarding stage, such as [11,17,7,29,19]. It is important to mention that these mechanisms consider that the Originator is static. Thus, the query results are always sent to the point where the queries were issued.

In [10] it is considered that the Originator can move during the query process. The proposed mechanism uses the same protocol used in the Forwarding stage (SQR). Nevertheless, it disseminates the query result to the nodes in a circular region where the Originator probably is. The center of this region is the point where the query was issued in the WSN and the radius is calculated multiplying the Originator speed by the time spent to process the query.

3.5.1. Analyzing the strategies for the Return stage

In window query processing, the use of the opposite route created in the Forwarding stage to return the query result depends on node mobility. In mobile WSN this strategy may not be possible, since nodes may move away and the route may be disrupted. However, in static WSN only node failures can compromise the use of this strategy. Another option is the creation of a new route using the same protocol of the Forwarding stage.

The data dissemination proposed by IWQE [30] and IKNN [19] creates itineraries in which the path starts and finishes in different nodes. In this case, it is not possible to use the opposite route created in the Forwarding stage. Hence, the mechanisms use GPSR, which is also used in the Forwarding stage.

3.6. Distributed spatial indexes

In traditional spatial databases, spatial indexing techniques are used to improve spatial query processing. These indexes are cre-

[2] Only IKNN disseminates by itineraries. The others mechanisms proposed in this work use other strategies.

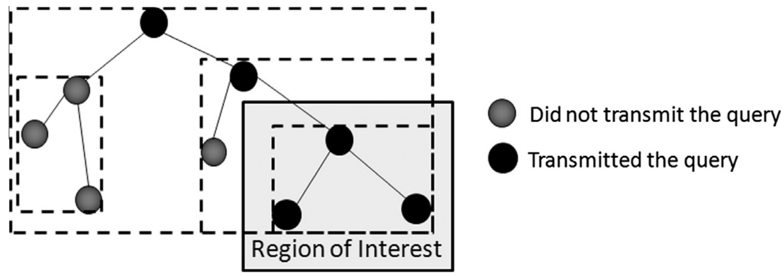

Fig. 10. Query Dissemination using Georouting Tree.

ated in a centralized manner. In WSN, however, queries are processed in a distributed manner. Moreover, WSN topologies can change over time, i.e. nodes fail due to energy depletion and mobility. Hence, traditional indexes cannot be applied to in WSN in an energy efficient manner [36].

Some works defined location-based distributed spatial indexes for spatial query processing in WSN in order to reduce the number of nodes that process spatial queries. The data that constitutes the indexes is distributed over the nodes. The index relieves the WSN to forward and disseminate the queries to the nodes inside the region of interest. In the following, we present an overview of spatial indexes in WSN for spatial query processing.

3.6.1. GRT – Georouting Tree

Nodes in a routing tree save the identifier of the node immediately superior to them in this tree (its parent). Hence, when a node needs to send a message to the sink, it sends this message to its parent, which will send the message to its parent too. This process is repeated until the root is reached. In GRT [37], each node saves who is its parent as well as a bounding box that covers itself and all its descendents. This information constitutes a spatial index that helps the forwarding of the queries. When a node receives a query, it verifies if the region of interest overlaps its bounding box. If so, it transmits the query to its children. If not, it drops the query. Hence, this strategy avoids sending the queries to all nodes in the WSN. Fig. 10 illustrates GRT. Only the black nodes transmitted the query.

3.6.2. SPIX – Spatial IndeX

SPIX is a mechanism developed for window queries [15]. It creates a spatial index, in order to simplify spatial query processing. This index is created over a routing tree having the sink as the root. In SPIX, each node stores a two-dimensional minimum bounding area (*MBA*). The MBA is a region that that covers the node and the MBAs of its descendents in the routing tree. When a node receives a query, it verifies if the region of interest of the query and its MBA overlap. If so, the node will participate in the query processing. SPIX tries to minimize the MBA of the nodes, because smaller MBAs will reduce the amount of nodes participating in query processing.

SPIX exploits two models for creating the MBA: rectangular and angular. In the *rectangular model*, the MBA is the minimum bounding rectangle (MBR) that covers the node and all nodes below it in the routing tree (similar to GRT). In the *angular model*, the MBA is the minimum bounding pie represented by a start/end radius and start/end angles. An evaluation using simulations showed that the best performance was reached by the rectangular model.

The minimization of the node's MBAs is called "energy optimization phase", and works as follows. If a node identifies that it is located inside two different MBAs, it runs a "parent-switching verification" process. This process checks if the MBA of the node's parent in the routing tree will be decreased if it changes to the other node's routing tree. Fig. 11 depicts a node that changes its parent to decrease the old parent's MBA.

3.6.3. Cluster and index

Park et al. [16] described a mechanism for window query processing that creates a semi-distributed spatial index to disseminate queries and to retrieve data. This mechanism divides the WSN in square sub-regions. Nodes in the same square constitute a cluster. The cluster-head is the node closest to the top right corner of each sub-region. Each cluster-head receives the location of all nodes in its cluster. Thus, the cluster-head knows the whole topology of its section. They run either Dijkstra's shortest path algorithm or Prim's minimum spanning tree algorithm to build their energy efficient local tree. Fig. 12 shows an area monitored by a WSN, their sub-regions, nodes and cluster-heads.

The mechanism uses an algorithm based on MBR, such as SPIX, to disseminate queries from the sink to the cluster-heads. After receiving a query, each cluster-head disseminates the query in its cluster, aggregates all data collected there and sends back the query result. This mechanism works in a centralized manner within the clusters and in a distributed manner among the clusters. This strategy allows the use of the best routes inside the cluster and avoids the high energy consumption of a totally centralized algorithm. However, it overloads the cluster-heads, since they centralize all communication in their clusters.

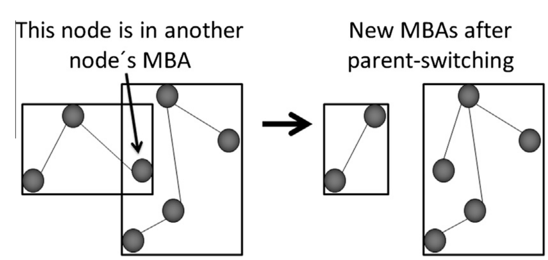

Fig. 11. Parent-switching verification phase.

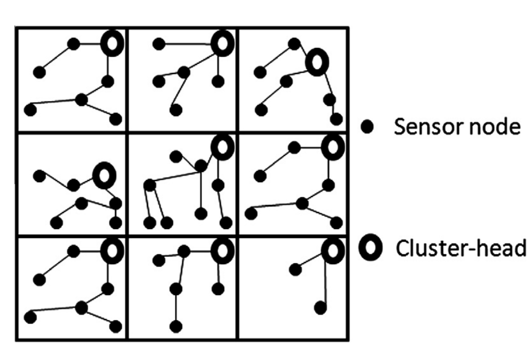

Fig. 12. Cluster in each subregion.

3.6.4. Back Forwarding method

Li et al. [8] described a mechanism also based on the routing tree created by SPIX. In SPIX, a node *A* adds its MBR to the MBR of its parent's candidate. Then, node *A* chooses as parent the node which will have the smallest new MBR. However, this mechanism does not guarantee the best aggregation process. In some cases, the query can reach the region of interest by more than one node. Fig. 13a shows an example. Thus, part of the aggregation can occur outside the region of interest, increasing the amount of message transmissions.

In the mechanism described in [8], the Forwarding stage uses MBR, but in the Aggregation stage a node chooses a different parent for each query. Nodes have a list of parent candidates. They prefer a parent that is within the region of interest, so the aggregation only occurs inside this region. This strategy saves a substantial part of the energy consumed in the Aggregation stage. Fig. 13b shows an example of aggregation limited to the region of interest.

3.6.5. Analyzing the distributed spatial indexes

The mechanisms found in the literature on distributed spatial indexes are based on minimum bounding rectangle. Hence, they have the same drawbacks of query processing. Queries can be started only in the sink, because the indexes are created based on the location it. The works [11,17,7,30,10] consider queries issued from any nodes in the network. Hence, this kind of index is not appropriated for this type of WSN.

Moreover, the WSN consume too much energy if nodes are mobile. If a node moves away, its parent's MBR will change and its children will look for a new parent. If a node receives a message informing that its children's MBR changed, it will update its MBR and notify its parent. Hence, the movement of a node can produce changes that are propagated until the sink. So, node mobility can provide several update messages and, consequently, consume too much energy. However, this type of index can be used in static WSN. The works [15,16,8] showed the energy saving of using spatial indexes in static WSN.

4. Spatial query processing mechanisms

This section presents an overview of the most relevant spatial query processing mechanisms for WSN found in the literature. We classify these mechanisms into window query and KNN query processing. The window query processing mechanisms are classified into structured (create a global structure such as index or cluster to process queries) and unstructured (do not create structures). All KNN mechanisms in the literature are unstructured.

Table 1 presents a summary of all mechanisms analyzed in this work. However, only those in boldface are described ahead. We did not include the stages Pre-Processing and Sensing. Pre-Processing is considered just in our previous works [11,17] and none of the analyzed works about spatial query processing consider the Sensing stage.

4.1. Window query processing

4.1.1. Structured mechanisms

Georouting Tree: it is the first spatial query mechanism that creates a global structure based on MBR. The drawback of this mechanism, and of the others based on the MBR, is that queries only start in the sink. Moreover, this mechanism does not optimize the index, such as more recent mechanisms.

SPIX – Spatial IndeX: the main contribution of this work is the optimization phase that decreases the nodes' MBR. It is the most referenced distributed index in the literature.

4.1.2. Unstructured mechanisms

Irregular shapes: considers WSN deployed on areas where a disaster has occurred. The main contribution of this mechanism was the energy efficient processing of queries with regions of interest having irregular shapes. A drawback of this work is the accuracy of the query result, which can be reduced with the execution of the Pre-Processing stage.

SWIP – Spatial Window Processing: the main contribution of this work is the study of Forwarding and Dissemination algorithms.

IWQE – Itinerary-base Window Query Execution: the main contribution of this mechanism is the definition of an structure-free window query processing technique based on itineraries [30]. A drawback of this technique is the strong dependency on the WSN topology.

4.1.3. Analyzing the window query processing mechanisms

Xu et al. [30] compared four window query mechanisms: Flooding, GeoRouting Tree, WSI (Window Spanning Infrastructure – forwards using GPSR and disseminates using Restricted Flooding) and IWQE. This last mechanism presented the best results, having query accuracy very close to Flooding. GRT presented the worst performance, since it suffers from node mobility. WSI presented an average performance, except in extreme conditions (high query load or large window area). These experiments showed that itinerary-based mechanisms are the most promising for window query. They can be used in static or mobile WSN and the query can be issued by any node. IWQE presents low energy consumption when compared with the other mechanisms. However, its accuracy depends on the WSN topology.

4.2. KNN query processing

GRT, KBT and IKNN: are three KNN processing mechanisms. The experiments showed that IKNN provides the smallest energy consumption, followed by GRT and KBT, respectively. The best query result accuracy was provided by GRT, mainly in sparse WSN.

PCIKNN – Parallel Concentric Itinerary KNN: this mechanism calculates the network density during the traversal of the itineraries, hence the radius of the region of interest can be adjusted to compensate for spatial irregularity and node mobility. The experiments

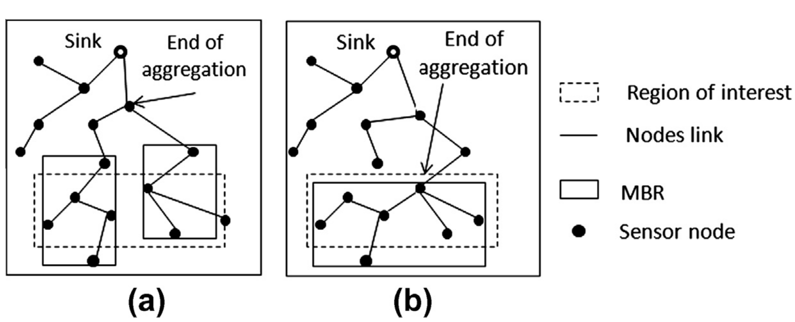

Fig. 13. (a) MBR created by SPIX and (b) MBR for a more efficient Aggregation.

showed that PCIKNN presents smaller energy consumption and query latency than similar mechanisms.

4.2.1. Analyzing the KNN query processing mechanisms

In [19], three mechanisms for KNN queries were compared. They are: GRT, KBT and IKNN. The experiments showed that IKNN substantially reduces the energy consumption. However, it will also reduce the accuracy of the query result and increase the query latency in the majority of the analyzed scenarios. GRT presented the best query accuracy, but its energy consumption and latency were the worst. KBT had a performance advantage in all the metrics except for energy consumption. The mechanisms DIKNN, IKNN and PCIKNN were compared in [20]. The parallel strategies for KNN query processing present better performance (energy consumption, query latency and query accuracy), since DIKNN and PCIKNN had better results. PCIKNN showed the best results in practically all analyzed scenarios.

5. Using information fusion to improve spatial queries

Spatial query mechanisms are a mean for requesting information from WSN, and are suitable for the use of information fusion techniques. Our review of the literature indicated that most works on spatial queries focus on the Forwarding and Dissemination stages, while the Sensing and Aggregation stages did not receive much attention. Information fusion algorithms could be used in these stages, where the sensor data is collected and summarized. As a consequence, the amount of packets generated during query processing can be reduced, and as such the energy consumption of the mechanisms decreases.

There is, however, a trade-off in the use of information fusion. In recent studies, it has been shown that the heavy use of inference and summarization techniques in WSN places a heavy burden on the mechanisms, since the loss of a data packet penalizes much more the precision of the network [35]. Without information fusion, high packet loss rates can be tolerated, because of the redundancies of sensor readings. When in-network aggregation occurs, on the other hand, the importance of each packet increases, and as such the mechanisms must spend more energy in an attempt to guarantee the correct delivery of all packets. Thus, redundancies may be desirable when applying information fusion techniques in sensor networks.

6. Research opportunities in spatial query processing

This section presents research opportunities in spatial query processing. There are network and environmental characteristics that were not considered by the mechanisms found in the literature. However, these characteristics can be found in real WSN applications. If these characteristics are considered, they generate several challenges for spatial query processing mechanisms.

Node mobility: mobility in WSN can be divided into node mobility and sink mobility. In the analyzed mechanisms, all nodes store information about their neighbor links or about distributed indexes. Node mobility changes the topology, thus producing changes in the indexes. In other mechanisms, nodes send periodic beacons to maintain a local view of the network. The beaconing period will depend on the amount of node mobility. Moreover, if the sink is mobile, the point that the query was issued may not be the point where the query result must be retrieved. No work presented a protocol to solve this problem in an energy efficient manner.[3]

[3] In [10] was proposed SQR to manage this problem, however it was not evaluated.

Table 1
Mechanisms for spatial query processing.

Name	Mobility		Query		Originator	Structured		Mechanism operation			
	Node	Sink	Type	Result		Index	Cluster	Fwd.	Diss.	Agg.	Ret.
Georouting Tree [37]	No	No	Window	Snapshot	Sink	Yes	No	MBR		Yes	MBR
SPIX [15]	No	No	Window	Snapshot	Sink	Yes	No	MBR		Yes	MBR
Cluster index* [16]	No	No	Window	Snapshot longlife	Sink	Yes	Yes	MBR		Yes	MBR
Back Forwarding* [8]	No	No	Window	Snapshot	Sink	Yes	No	MBR		Yes	Optimized MBR
Irregular shapes* [11,17]	No	No	Window	Snapshot	Any node	No	No	Greedy	Restricted	Yes	Created by greedy
SWIP [7]	No	Yes	Window	Spatio-temporal	Any node	No	No	Greedy/Flooding	WinDepth/Restricted	Yes	Created in Forwarding
IWQE [30]	Yes	No	Window	Snapshot	Sink	No	No	GPSR	Itinerary	Yes	GPSR
SWOP [38]	No	No	Window	Snapshot	Sink	No	Yes	N/A	N/A	No	N/A
Extension of TinyDB* [29]	No	No	Window	Spatio-temporal	Sink	No	No	Flooding		Yes	Created by Flooding
Highly dynamic* [10]	Yes	Yes	Window	Snapshot	Any node	No	Yes	SQR	SPA	Yes	SQR
MobiQuery [9]	Yes	Yes	Window	Spatio-temporal	Sink	No	No	No	Restricted	Yes	No
GRT [19]	Yes	No	KNN	Snapshot	Any node	No	No	Flooding		Yes	Created by Flooding
KBT [19]	Yes	No	KNN	Snapshot	Any node	No	No	GPSR	Restricted	Yes	GPSR
IKNN [19]	Yes	No	KNN	Snapshot	Any node	No	No	GPSR	Itinerary	Yes	GPSR
DIKNN [18]	Yes	No	KNN	Snapshot	Sink	No	No	GPSR	Itinerary	Yes	GPSR
PCIKNN [20]	Yes	Yes	KNN	Snapshot	Any node	No	No	GPSR	Itinerary	Yes	GPSR

* The original paper does not propose a name, this name is based on the mechanism's characteristics.

Delay-tolerant Wireless Sensor Networks (DTWSN): is a novel network paradigm that can provide communication services in scenario where persistent network connectivity is not available. In such scenarios, sensors nodes are sparsely deployed or are moving on large regions [39]. All spatial query processing stages must consider these characteristics to process the query. This kind of WSN provides several challenges, such as routing, dissemination and aggregation in WSN with intermittent connected nodes.

Cross-layer design: the analyzed works propose spatial query processing mechanisms that work in the routing layer. No mechanism uses information from other layers in order to improve their operation. An example is the use of signal strength (from the physical layer), which can help choosing the best node to forward queries and query results. Inumerous works showed that routing over wireless links must consider link reliability. This metric significantly increases the reliability and the throughput of the flows [40]. The same should be done for the Forwarding and Return stages of spatial query processing.

Security: WSN can be attacked by malicious nodes in order to obtain confidential data or disturb the WSN operation. Hence, spatial query processing mechanisms must identify the location of malicious nodes and perform algorithms to maintain the WSN integrity and data security. No work analyzed here studies the security issues of spatial query processing.

Heterogeneous WSN: the works on spatial query processing consider homogeneous nodes. However, WSN can be formed by different kinds of sensor nodes. Spatial query processing mechanisms should manage the heterogeneity in order to reduce resources usage.

Collaborative WSN: several WSN can collaborate in order to improve spatial query processing. The works analyzed here consider only one WSN over the monitored region. However, many WSN can be deployed in the same region and work together in query processing.

Sensing stage: existing works consider neither energy consumption in the Sensing stage nor strategies to reduce it. In several applications, such as environmental monitoring, nodes close to each other will sense similar information. Thus, information fusion techniques could be employed to reduce the number of messages transmitted. The use of those techniques in spatial queries has been neglected in previous works.

Density control: most of the related works do not consider nodes in sleep mode during the query processing. However, nodes in sleep mode can decrease the accuracy of the query results. In KNN queries, the K nodes that collected data may not be the nearest nodes of the query point. In window queries, aggregation operations such as *max* and *min* may not retrieve the correct value.

7. Conclusion

This work presented a state-of-the-art on spatial query processing in WSN. We divided the processing in six stages and analyzed how each related work performs them. This division made easier to understand how each mechanism performs the query processing. Moreover, it allowed the classification of these mechanisms and the queries processed by them.

The first stage is called Pre-Processing. It is performed in the users' computer, preparing the queries to be processed. In the Forwarding stage, location-based routing protocols forward the queries from the Originator to the region of interest. Most of the mechanisms use GPSR because it has low energy consumption and the number of dropped queries is small. In the Dissemination stage, itinerary-based protocols provide the best energy consumption. However, these protocols are influenced by the itinerary formats and WSN topologies. The Sensing stage was not studied by any related work, since its operation is dependent on the application. In the Aggregation stage, several strategies proposed in the literature reduce substantially the energy consumption. However, they depend on the protocol used to disseminate queries. Finally, in the Return stage, the mechanisms also can use several routing protocols, but most of them use GPSR.

The queries found in the literature were classified according dissemination and query result. Dissemination specifies which nodes will answer the query and defines two types of queries: window and KNN. The query result specifies when nodes will send their collected data. We found snapshot, spatiotemporal and long-life queries. We classified the spatial query processing mechanisms in unstructured and structured. In the unstructured mechanisms, nodes know only the location of their neighbors. In the structured mechanisms, distributed indexes are created in WSN in order to help processing queries.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Commun. 38 (2002) 393–422.

[2] M. Demirbas, X. Lu, Distributed quad-tree for spatial querying in wireless sensor networks, in: IEEE International Conference on Communications (ICC), Glasgow, Scotland, pp. 3325–3332.

[3] S.R. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, TinyDB: an acquisitional query processing system for sensor networks, ACM Trans. Database Syst. 30 (2005) 122–173.

[4] J. Wang, R.K. Ghosh, S.K. Das, A survey on sensor localization, IET Control Theory Appl. 8 (2010) 2–11.

[5] Y. Manolopoulos, A. Papadopoulos, M. Vassilakopoulos (Eds.), Spatial Databases: Technologies, Techniques and Trends, 1st ed., Idea Group Publishing, 2005.

[6] M. Sharifzadeh, C. Shahabi, Supporting spatial aggregation in sensor network databases, in: 12th Annual ACM International Workshop on Geographic Information Systems (GIS), ACM, 2004, pp. 166–175.

[7] A. Coman, J. Sander, M.A. Nascimento, Adaptive processing of historical spatial range queries in peer-to-peer sensor networks, Distrib. Parallel Dat. 22 (2007) 133–163.

[8] Y. Li, S.-H. Eo, D.-W. Lee, Y.-H. Oh, H.-Y. Bae, An energy efficient adaptive back forwarding method for spatial range query processing in sensor networks, in: International Conference on Advanced Language Processing and Web Information Technology (ALPIT), IEEE Computer Society, 2008, pp. 373–379.

[9] C. Lu, G. Xing, O. Chipara, C.-L. Fok, S. Bhattacharya, A spatiotemporal query service for mobile users in sensor networks, in: 25th IEEE International Conference on Distributed Computing Systems (ICDCS), IEEE Computer Society, 2005, pp. 381–390.

[10] Y. Xu, W.-C. Lee, Window query processing in highly dynamic geosensor networks: issues and solutions, in: A. Stefanidis, S. Nittel (Eds.), GeoSensor, Networks, 2004, pp. 31–52.

[11] R.I. da Silva, V. del Duca Almeida, A.M. Poersch, J.M.S. Nogueira, Spatial query processing in wireless sensor network for disaster management, in: 2nd IFIP Wireless Days, pp. 1–5.

[12] IEEE, Wireless MAC and PHY Specifications for Low Rate WPAN, IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2006), IEEE, vol. 1, 2006, pp. 1–305.

[13] A. Deshpande, Z. Ives, V. Raman, Adaptive query processing, Found. Trends Databases 1 (2007) 1–140.

[14] Z. Jin, Y. Jian-Ping, Z. Si-Wang, L. Ya-Ping, L. Guang, A survey on position-based routing algorithms in wireless sensor networks, Algorithms 2 (2009) 158–182.

[15] A. Soheili, V. Kalogeraki, D. Gunopulos, et al., Spatial queries in sensor networks, in: 3th Annual ACM International Workshop on Geographic Information Systems (GIS), ACM, 2005, pp. 61–70.

[16] K. Park, B. Lee, R. Elmasri, Energy efficient spatial query processing in wireless sensor networks, in: 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW), IEEE Computer Society, 2007, pp. 719–724.

[17] R. Silva, V. Almeida, A. Poersch, J.-M.S. Nogueira, Wireless sensor network for disaster management, in: 12th IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 870–873.

[18] S.-H. Wu, K.-T. Chuang, C.-M. Chen, M.-S. Chen, DIKNN: an itinerary-based KNN query processing algorithm for mobile sensor networks, in: International Conference on Data Engineering (ICDE), pp. 456–465.

[19] Y. Xu, T.-Y. Fu, W.-C. Lee, J. Winter, Processing K nearest neighbor queries in location-aware sensor networks, Signal Process. 87 (2007) 2861–2881.

[20] T.-Y. Fu, W.-C. Peng, W.-C. Lee, Parallelizing itinerary-based KNN query processing in wireless sensor networks, IEEE Trans. Knowl. Data Eng. 22 (2010) 711–729.

[21] M. Baldauf, S. Dustdar, F. Rosenberg, A survey on context-aware systems, Int. J. Ad Hoc Ubiquitous Comput. 2 (2007) 263–277.

[22] D.F. Macedo, A.L. dos Santos, J.M.S. Nogueira, G. Pujolle, A distributed information repository for autonomic context-aware MANETs, IEEE Trans. Netw. Serv. Manage. 6 (2009) 45–55.

[23] J. Gao, L. Guibas, N. Milosavljevic, J. Hershberger, Sparse data aggregation in sensor networks, in: 6th International Conference on Information Processing in Sensor Networks (IPSN), ACM, 2007, pp. 430–439.

[24] X. Zhou, G. Xue, C. Qian, M. Li, Efficient data suppression for wireless sensor networks, in: 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS), IEEE Computer Society, 2008, pp. 599–606.

[25] M. Nelson, J.-L. Gailly, The Data Compression Book, second ed., John Wiley & Sons, 1995.

[26] R. Wu, M. Chen, Y. Su, H.J. Siddiqui, A novel location-based routing algorithm for energy balance in wireless sensor networks, in: International Conference on Communications and Mobile Computing (CMC), IEEE Computer Society, 2009, pp. 568–572.

[27] Y. Xu, W.-C. Lee, J. Xu, G. Mitchell, Energy-aware and time-critical geo-routing in wireless sensor networks, Int. J. Distrib. Sens. Netw. 4 (2008) 315–346.

[28] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, Ad Hoc Netw. 3 (2005) 325–349.

[29] L.P. Paolino Di Felice, Massimo Ianni, A spatial extension of TinyDB for wireless sensor networks, in: IEEE Symposium on Computers and Communications (ISCC), pp. 1076–1082.

[30] Y. Xu, W. Chien Lee, J. Xu, G. Mitchell, Processing window queries in wireless sensor networks, in: 22th IEEE International Conference on Data Engineering (ICDE), 2006, p. 70.

[31] F. Zhao, L.J. Guibas, Wireless sensor networks: an information processing approach, The Morgan Kaufmann Series in Networking, first ed., vol. 1, Morgan Kaufmann Publishing, 2004.

[32] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: 6th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 243–254.

[33] R. Finkel, J.L. Bentley, Quad trees: a data structure for retrieval on composite keys, Acta Inform. 4 (1974) 1–9.

[34] E.F. Nakamura, A.A.F. Loureiro, A.C. Frery, Information fusion for wireless sensor networks: methods, models, and classifications, ACM Comput. Surv. 39 (2007) 1–55.

[35] U. Raza, A. Camerra, A.L. Murphy, T. Palpanas, G.P. Picco, What does model-driven data acquisition really achieve in wireless sensor networks?, in: IEEE International Conference on Pervasive Computing and Communication (PerCom), Switzerland.

[36] M.A. Casanova, G. Câmara, C.A. Davis, L. Vinhas, G.R. de Queiroz, Banco de Dados Geográfico, MundoGeo, Curitiba, first ed., vol. 1 of 1, 2005.

[37] D. Goldin, M. Song, A. Kutlu, H. Gao, H. Dave, Georouting and delta-gathering: Efficient data propagation techniques for geosensor networks, in: First Workshop on Geo Sensor, Networks, pp. 9–11.

[38] G. Jin, S. Nittel, Toward spatial window queries over continuous phenomena in sensor networks, IEEE Trans. Parallel Distrib. Syst. 19 (2008) 559–571.

[39] A. Nayebi, H. Sarbazi-Azad, G. Karlsson, Routing, data gathering, and neighbor discovery in delay-tolerant wireless sensor networks, in: 24th International Symposium on Parallel & Distributed Processing (IPDPS), IEEE Computer Society, Washington, DC, USA, 2009, pp. 1–6.

[40] M.E. Campista, P.M. Esposito, I.M. Moraes, L.H.M.K. Costa, O.C.M.B. Duarte, D.G. Passos, C.V. de Albuquerque, D.C.M. Saade, M.G. Rubinstein, Routing metrics and protocols for wireless mesh networks, IEEE Netw. 22 (2008) 6–12.