Crisis Management using Spatial Query Processing in Wireless Sensor Networks

Mohammad Shakeri*
Department of Computer, Neyshabur Branch, Islamic Azad university, Khorasan Razavi, Iran
alborz.corp@gmail.com
Seyed Majid Mazinani
Department of Electrical Engineering, Imamreza International University, Mashhad, Iran
smajidmazinani@imamreza.ac.ir

Received: 04/Jan/2017 Revised: 11/Jun/2017 Accepted: 12/Jul/2017

Abstract

Natural disasters are an inevitable part of the world that we inhabit. Human casualties and financial losses are concomitants of these natural disasters. However, by an efficient crisis management program, we can minimize their physical and social damages. The real challenge in crisis management is the inability to timely receive the information from the stricken areas. Technology has come to the aid of crisis management programs to help find an answer to the problem. One of these technologies is wireless sensor network. With recent advances in this field, sensor nodes can independently respond to the queries from the users. This has transformed the processing of the queries into one of the most useful chapters in sensor networks. Without requiring any infrastructure, the sensor network can easily be deployed in the stricken area. And with the help of spatial query processing, it can easily provide managers with the latest information. The main problem, however, is the irregular shape of the area. Since these areas require many points to present them, the transmission of the coordinates by sensor nodes necessitates an increase in the number of data packet transmissions in the sensor network. The high number of packets considerably increases energy consumption. In related previous works, to solve this problem, line simplification algorithm s, such as Ramer-Douglas-Peucker (RDP), were used. These algorithms could lessen energy consumption by reducing the number of points in the shape of the area. In this article, we present a new algorithm to simplify packet shapes which can reduce more points with more accuracy. This results in decreasing the number of transmitted packets in the network, the concomitant reduction of energy consumption, and, finally, increasing network lifetime. Our proposed method was implemented in different scenarios and could on average reduce network's energy consumption by 72.3%, while it caused only 4.5% carelessness which, when compared to previous methods, showed a far better performance.

Keywords: Wireless Sensor Network; Query Processing; Spatial Query; Crisis Management; Image Processing.

1. Introduction

The world that we inhabit constantly witnesses the occurrence of unfortunate events, affecting countless people who are desperate to receive help. However, the measures taken immediately after these events are vital and play a determining role in the amount of fatalities and financial losses. One of these measures is crisis management. Crisis management refers to all the actions taken to minimize the physical and social impacts of an event. Moreover, latest information on disaster-stricken areas is necessary for decision-making processes and all the institutions involved in crisis management procedures. In disaster-stricken areas network and communication infrastructures are destroyed while normal social and economic interactions are impossible. As mentioned earlier, a proper understanding of the stricken area is necessary. At present, GPS, maps, and satellite images are used [1]. However, creating a sensing structure with high capabilities to predict dangers can prove very effective.

This structure must be quick and easy to launch, independent, and should not require any particular infrastructure. What is more, it should be adaptable to different conditions [2]. We propose using wireless sensor network which has all these features, and can fly over the stricken region and collect the latest information and transmit it to base station [3, 4]. Wireless sensor network consists of many small devices which sense, process, and save the environmental data [5]. These devices have limited power supplies and usually battery replacement is infeasible since the sensor network may be set to watch over rough regions [6]. As a mature technology, wireless sensor network has evolved into an alternate data gathering device. Currently, these devices are able to answer to query requests from users [7]. Some sensor network models work as distributed data station such as TinyDB. They process query requests similar to SQL which is an easy way for users to receive data from the region in which the network is installed [8].

With respect to the fact that Iran is in a very seismic area, conducting research in this field and dealing with natural disasters, especially crisis management, is of great import. A real challenge in crisis management is the immediate access to information related to the stricken area. This is the motivation behind the present article: to collate information from across the stricken area during the early hours by means of wireless sensor networks. Regarding the features of sensor networks, this network

* Corresponding Author

can be installed in the area during the early hours and provide involved institutions with valuable information.

When we intend to implement wireless sensor networks in disaster-stricken areas to collect a region's data to assist crisis management, we encounter a challenge. This challenge takes the form of the topology of the affected area. If the topological shape is a regular quadrilateral (Fig.2), then, using two points, the affected area can easily be specified for the sensor nodes. However, the topology of these areas is seldom regular; therefore, these irregular areas require a lot of points to be presented (Fig.3). Transmitting the coordinates of these points to sensor nodes necessitates increasing transmitted packets in the sensor network. The high rate of sending packets increases network's energy dissipation, particularly in multiple transmissions. Benefiting from line simplification algorithms, previous researchers were able to reduce the number of necessary points for the presentation of the region while at the same time they manages to reduce sensor network's energy consumption. In this article, we try to present a new simplification algorithm which reduces the points of the topology of the area more than previous works so as to have lesser energy consumption and carelessness.

The main objective of this article is to present an efficient algorithm to simplify packet irregular shapes so that it will be able to improve its capacity to function well in usages where either the communicative costs to determine irregular shapes are high or there is a severe communicative limitation (such as the one in wireless sensor networks). Inspired by regular shapes which require two points to be presented, the proposed algorithm puts forward a shape similar to the original one by using fewer points. These points demand less memory, less time, and fewer communicative packets to carry out transmission. In almost all former researches, line simplification algorithms were used to carry out shape simplification. However, our proposed algorithm is specifically designed for packet shapes and achieved better results than former algorithms.

In sections 2 and 3, processing spatial query requests and the topology of the affected area are investigated. In section 4, the review of related works will be presented. In section 5, we will explain the proposed algorithm. The implementation and the analysis of the results are included in section 6. Finally, part 7 deals with conclusion and suggestions for further research.

2. Spatial Query Processing in WSN

The wireless sensor network is installed and we intend to use the data collected by the network. We send our request for data to the sensor network, and the network responds to it through a procedure. This procedure is referred to as processing the queries (Fig. 1).

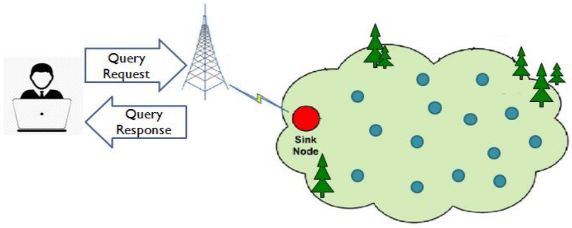

Fig. 1: Query processing in wireless sensor networks.[5]

The user requests his needed information in the form of some query requests from the base station. Queries will reach the nodes from the base station by means of a path which is calculated by a routing protocol, while processing will be carried out on queries and the intended data will be collected by the sensor nodes. Finally, query results will return to the base station via a return path [9]. The structure of a query may resemble SQL query with the following blocks [10]:

SELECT - FROM - WHERE - GROUPBY-HAVING

For example:

SELECT *temperature FROM sensors WHERE location = (x , y)*

If in a query a certain region or certain coordinates are specified for the nodes of the sensor network, and if the nodes can respond to the queries with regard to that region, then the query is an spatial one [11]. In this query, the user needs the information of only a certain part of the region in which the network is installed and tries to introduce the intended region to the sensor nodes by benefiting from such methods as specifying the list of nodes' coordinate, specifying the geometrical shapes such as circle, quadrilateral, polygon, and, finally, by specifying the trajectory path. The majority of mechanisms prepare a window for the nodes and only the nodes inside the windows respond to them. In some mechanisms a node is specified while the neighbouring nodes engage in responding to the queries. These sorts of queries are widely used in real situations [2].

3. Examining the Topology of Affected Area

Firstly, we should notify the nodes of the topology of the affected area. As it was mentioned earlier, we can easily specify the intended area by means of only two points for regular shapes such as squares and rectangles, meaning that only by sending one packet to sensor nodes, it is possible to introduce the area in the queries. As shown in Fig. 2, by sending a point with the lowest x,y as point A, and by sending another point with the highest x, y as point B, we can specify the entire intended area.

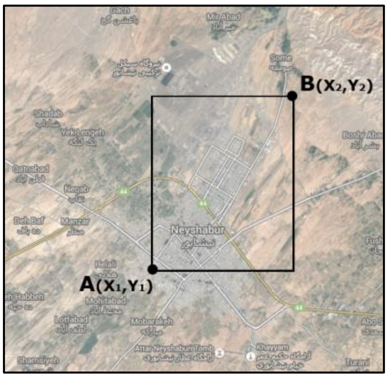

Fig. 2: Affected area with a regular shape [12]

As Fig. 2 illustrates, a sensor node whose coordinates are between the two points, is positioned within the intended area. Otherwise, it is situated beyond that area and should not respond to the queries. If a sensor node meets the following two conditions, it can respond to the queries:

$X_1 \leq \text{Node.x} \geq X_1$, $Y_1 \leq \text{Node.y} \geq Y_1$

However, this is quite unlike the irregular shapes where it is impossible to determine them with only two points since they require many. This is illustrated in Fig. 3.

Fig. 3: Affected area with irregular shape

In Fig. 3, the specified area is the city of Neyshabur. As shown in Fig. 3, the intended area is of an irregular shape and it is not possible to determine the shape of this area for the sensor nodes with only two points. For example, to determine the shape of this area we need 251 points. If each point requires two bytes, then we need 502 bytes of free memory space. And if the network packets are 28 bytes, then we need 18 packets to transfer the points which determine the shape of the area. Because of high energy consumption in wireless communications between the nodes of sensor networks, and multiple transmissions, this is very costly.

4. Related Works

Simplification algorithms are used to solve the problem of the number of points in irregular shapes. In older researches the Nth point method was used in such a way that the value of N is chosen by the user and the algorithm eliminates N points [13]. Or N points will be randomly eliminated [14]. But these methods are not accurate since they eliminate the points irrespective of their coordinates in the shape [15], and in the majority of cases the shape loses its essence after simplification.

One of the optimized algorithms that simplifies the lines logically is the Ramer-Douglas-Peucker algorithm which was presented by David Douglas and Thomas Peucker [16], and is the continuation of Urs Ramer's work [17]. This algorithm is tasked with finding a line segment which resembles the main line segment but with fewer points. By taking into account the beginning and the end of the line segment, this algorithm determines the farthest point from the line and divides it into two new line segments: one line segment from the starting point to the farthest point and one from the farthest point to the line end point. This operation is carried out for these two line segments separately.

A threshold is considered for this algorithm. If the distance of the farthest point from the line is more that the threshold, the line breaks into two line segments. But if it is less than the threshold value, that point can be eliminated. Consequently, it is possible to reduce the number of line points. Obviously, the threshold determines precision. The larger the threshold is, the more the eliminated points are. However, the line segment obtained from the algorithm bears little resemblance to the main line segment. If the threshold is small, line segment bears more resemblance but fewer points are eliminated. In the following, we will explicate the algorithm by providing an example. Take Fig. 4 as an example. We apply RDP algorithm to two AZB and AWB lines. For AZB line, the beginning and the end of the line segment, that is, line AB, are connected by a straight line. Now, we should calculate the distance from point Z to the line. The distance of point $Z = (x_z, y_z)$ and line AB by $(x_A, y_A), (x_B, y_B)$ is obtained from the following formula

$$d(Z, AB) = \sqrt{\left(x_z - x_A - \lambda_q(x_B - x_A)\right)^2 + \left(y_z - y_A - \lambda_q(y_B - y_A)\right)^2}$$

where λ_q is equal to

$$\lambda_q = \frac{(x_B - x_A)(x_z - x_A) + (y_B - y_A)(y_z - y_A)}{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Considering the threshold value, we can observe that the distance $d(Z, AB)$ of point $Z = (x_z, y_z)$ is above the threshold. Therefore we cannot eliminate the Z point and no simplification takes place.

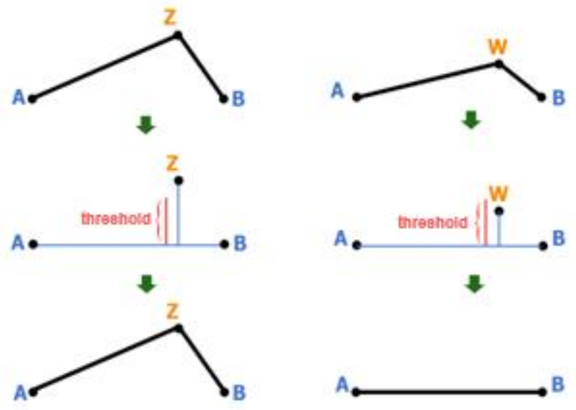

Fig. 4: RDP algorithm performance

Now we apply the algorithm to AWB line. First, the AB line is considered and then the distance from W to this line is calculated. This distance is below the threshold so it can be eliminated based on the algorithm's principles. The result is the AB line and simplification took place.

This algorithm can examine irregular shapes in a linear fashion and reduce its points. In the following, we will present some researches in which RDP algorithm was employed.

In [18] RDP algorithm is used in the fields of meteorology and geography in order to simulate the current of rivers and lakes and to draw the geographical maps.

In [19], the RDP simplification algorithm is utilised in navigation and routing systems so that they can carry out the best planning for linear paths with respect to optimum decision-making processes for telephone communication in navigation systems.

In [20], RDP has come to the aid of cultural heritage and architecture and has been able to classify Iberian ceramics with the help of simplified curves.

In [21], RDP has been used in image processing, and the utilised maps were drawn in robotic and industrial machines.

In [22], da Silva et al., used RDP algorithm in sensor networks and were able to simplify the area under examination to be used in Duty cycle mechanism in wireless sensor networks.

In [23], RDP simplification algorithm has been used in modelling and simulation, and a model to simulate evacuation in areas with complex paths is presented.

In [24], the researchers used RDP algorithm to simplify the queried area to reduce the points of the shape of the area. In this research, the output parameters of wireless sensor network simulation are presented using RDP simplification algorithm.

In [25], RDP algorithm is used to process images to automatically produce similar polygamies by specifying the threshold.

The researchers in [26], who deal with introducing and examining spatial query processing mechanisms, use RDP algorithm in pre-processing phase to simplify the queried area.

In [27], with regard to Douglas algorithm, large data collected across the route by position-based systems are reduced by simplification algorithms, and processing volume in position-based systems is considerably improved and optimized.

In [28], with the help of genetic and Douglas simplification algorithms, 3D printing programs are optimized.

5. The Proposed Algorithm to Simplify the Shape of the Area

The idea for this algorithm is derived the regular quadrilateral and rectangular shapes. The fact that we require two points to present regular shapes led us into using an algorithm which benefits from this feature. In this method, as opposed to other algorithms to which the irregular shape appears as a line and try to simplify the points on the line.

Points forming the shape of the irregular area is defined by IMG: $IMG = \{PI_1, PI_2, PI_3, \dots, PI_n\}$

where PI_1 is one of the points forming the shape of the area and $PI_i = \left(x_{PI_i}, y_{PI_i}\right)$.

we have processed the irregular shapes as closed ones. With respect to the nature of quadrilateral shapes which require two points to be presented, the proposed method divides the whole area into equal smaller quadrilaterals. To do this, first we need to determine the entire quadrilateral span of the irregular shape.

In order to do this, we should find the leftmost, rightmost, topmost, and bottommost point of the irregular shape. In other words, we look for two points which have the smallest and the biggest x, and two points which have the smallest and the biggest y, similar to Fig. 5.

Determining the leftmost and rightmost, the topmost and bottommost point in the shape of the area to form the regular frame of the shape:

a is the leftmost, b is the rightmost, c is the bottommost, and d is the topmost point

$\{a, b, c, d\} \in IMG$

$$a = (x_a, y_a) = \{PI_i | PI_i \in IMG\ , x_{PI_i} = \min\left(x_{PI_i} \in IMG\right)\}$$
$$b = (x_b, y_b) = \{PI_i | PI_i \in IMG\ , x_{PI_i} = \max\left(x_{PI_i} \in IMG\right)\}$$
$$c = (x_c, y_c) = \{PI_i | PI_i \in IMG\ , y_{PI_i} = \min\left(y_{PI_i} \in IMG\right)\}$$
$$d = (x_d, y_d) = \{PI_i | PI_i \in IMG\ , y_{PI_i} = \max\left(y_{PI_i} \in IMG\right)\}$$

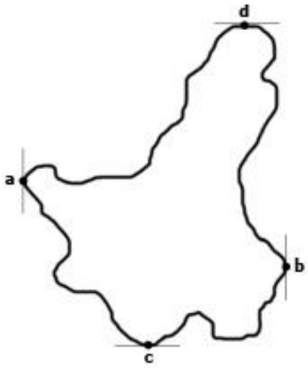

Fig. 5: Identifying determining points of the area

Using these four points, the bottom right and top left corners can be calculated. To calculate the x,y of point A, which is situated at the bottom left corner, we should use the x of point a and the y of point c, and to identify the x,y of point B, which is located at the top right corner, we need to use the x of point b and the y of point d. This is illustrated in Fig. 6.

A is the smallest point of the span and $A = (x_a, y_c)$
B is the biggest point of the span and $B = (x_b, y_d)$
The size of two horizontal sides is shown by LX and
$LX = x_b - x_a$
The size of two horizontal sides is shown by LY and
$LY = y_b - y_a$

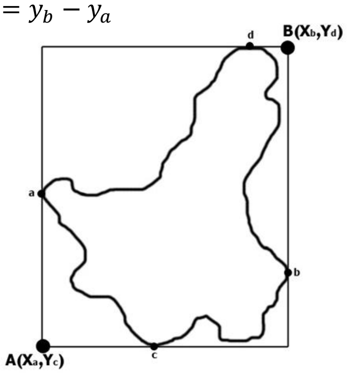

Fig. 6: Determining the span of the shape of the area

Now, by calculating points A and B, we can proceed with dividing the quadrilaterals. We should divide the entire area covered by the quadrilateral extending from point A to point B into quadrilaterals with equal length. Fig. 7 illustrates this point.{Tobler, 1964 #39}

The size of the threshold is shown by t which is equal to the size of quadrilaterals' sides, but must be adjusted according to the size of the frame so that quadrilaterals can cover the entire span. Determining tx which shows the size of horizontal and parallel sides to x axis $tx = \frac{LX}{[^{LX}/_{t}]}$

Determining ty which shows the size of horizontal and parallel sides to y axis $ty = \frac{LY}{[^{LY}/_{t}]}$

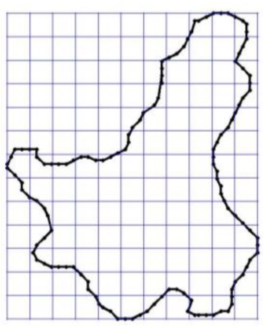

Fig. 7: Quadrilateral divisions in the proposed algorithm

After the division, the coordinates of the four corners of each quadrilateral is recorded.

Each quadrilateral is shown by determining each of its four corners $QL_i = \{Pdl_i, Pdr_i, Pul_i, Pur_i\}$

Pdl is the bottom-left corner of the quadrilateral

$$Pdl_i = \{\, Pdl_i = \left(x_{Pdl_i}, y_{Pdl_i}\right) | x_{Pdl_i} = x_{Pdl_i} + (hx \times tx)$$
$$, y_{Pdl_i} = y_{Pdl_i} + (hy \times ty)\,,$$
$$hx \in \{1,2,\dots,([^{LX}/_{t}] - 1)\},$$
$$hy \in \{1,2,\dots,([^{LY}/_{t}] - 1\})\}$$

The total number of quadrilaterals within the area A and B reaches 143. Some of these are situated within the shape, and some are situated outside the shape. Then, the quadrilaterals which are inside the shape must be determined. Even if only center of quadrilateral fall inside the shape, we should consider that quadrilateral as entirely falling within the shape.

To determine whether a point is inside or outside of the shape, we used a simple method: we move in four directions: right, left, top, and bottom. If in all these four directions we come across a point in the irregular shape, that point is very probably inside the irregular shape. However, if we do not come across in any of these four directions, the intended point is outside the shape. In other words, between the points of the irregular shape there must be four other points two of which must have an ***x*** equal to that of the intended point; and one of their ***y***s must be smaller and the other must be greater than the *y* of the intended point. The other two points of the irregular shape must have a ***y*** equal to that of the intended point, while one of their ***x***s must be smaller and the other must be greater than the intended ***x***. if these four points with these features are found among the points of the irregular shape, the point is very probably inside the shape. This means that given the ***x*** and ***y*** of the intended point, it must be looked for among the points of the irregular shape so that we can find four points possessing the previously mentioned features. The condition for being inside is that the center of the quadrilateral must lie inside shape.

First, we define an entire set of points on or inside the shape as Pin The four intended points are defined as $\{P_j, P_k, P_t, P_r\} \subset IMG$

and

$$Pin = \{P_i | A \le P_i \le B\, , x_{P_i} = x_{P_j} = x_{P_k}\, ,$$
$$y_{P_i} \ge y_{P_j}, y_{P_i} \le y_{P_k}\, , y_{P_i} = y_{P_t} = y_{P_r}\, ,$$
$$x_{P_i} \ge x_{P_t}, x_{P_i} \le x_{P_r} \;\; | \; \{P_j, P_k, P_t, P_r\} \subset IMG \,\}$$

Now we should say that if the center of the quadrilateral is either inside or on the shape, it is determined as being inside. In other words, the center of the quadrilateral must be a member of the set Pin.

The quadrilateral which lies inside the shape is defined by $QLin$ set:

$$Qlin = \{QL_i | (\left[QL_{\substack{x \\ Pdl \\ i}} + \frac{tx}{2}\right], \left[QL_{\substack{y \\ Pdl \\ i}} + \frac{ty}{2}\right]) \in Pin\}$$

For example, in Fig. 8 we can observe the quadrilaterals which are considered as falling within the shape. Out of 143 quadrilaterals, only 83 fall within the shape. Accordingly, we could manage to turn irregular shapes into regular quadrilateral shapes.

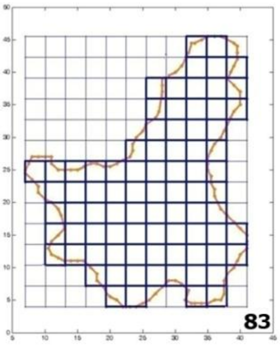

Fig. 8: Determining the quadrilaterals inside the shape using the proposed algorithm

As illustrated in the above Figure, the smaller the quadrilaterals become, the more their presenting accuracy increases. Therefore, the size of these quadrilaterals can act as the threshold that determines accuracy. The smaller the size of the threshold becomes, the more accuracy and the number of quadrilaterals increase. And to the extent the threshold gets bigger, both accuracy and the number of quadrilaterals decrease.

As it was mentioned before, to specify each quadrilateral, we require two points. As Figure 8 illustrated, 83 quadrilaterals are inside the shape; consequently, the total number of points which indicate the irregular shape are 164. If the quadrilaterals shrink in size, then certainly the number of points will increase. The points required to present the unsimplified irregular shape is 251. As a result, we could considerably reduce the number of points necessary to present the irregular shape. It should be noted, however, that, as with other simplification algorithms, the reduction in the number of points is inaccurate, meaning that some parts from the original shape are eliminated while a part is annexed to it which did not previously belong.

In the following, by merging the quadrilaterals which are inside the shape, we try to reduce the number of points even more that in simplification algorithm. The strategy of the merge algorithm is to merge the quadrilaterals that are inside the shape, and whose points have been discretely presented, into a bigger one so that all these quadrilaterals and their points change into one quadrilateral with only one point.

$$Qlmerg = \{\sum_{i=i+tx}^{LX} QLin_i \mid QLin_{\substack{x \\ Pdl \\ i}} = QLin_{\substack{x \\ Pdl \\ i+1}}\}$$

As it is obvious from Fig. 9, several of the quadrilaterals that are inside the shape and are attached to one another can be specified as a bigger quadrilateral. Thus, any number of quadrilaterals which are merged are possible to be presented by two points only. The more quadrilaterals we are able to merge, therefore, the more we reduce the number of points of a shape.

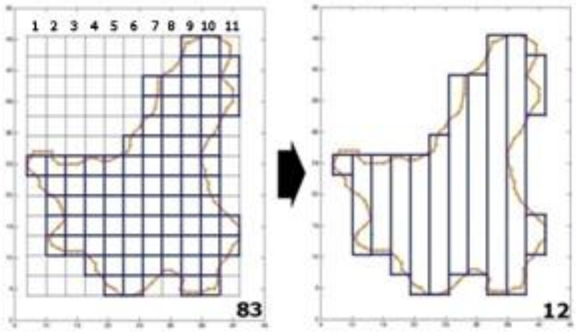

Fig. 9: Columnar merge in the proposed algorithm

The proposed algorithm merges quadrilaterals in a column. As the Figure clearly shows, we merge quadrilaterals which are inside the shape and are in the same column. In other words, it is possible to present the quadrilaterals which enjoy equal x and are situated in one column with two points. The most desirable condition occurred in column 9 where 13 quadrilaterals were merged and formed a bigger one. In place of transmitting 26 points to present 13 quadrilaterals, now it is possible to transmit only two points. However, the worst condition occurred in column 1. Apparently, we did not manage to merge any quadrilaterals in this column and, therefore, we did not experience any reduction in the number of points. If all the quadrilaterals are attached to one another in one column, then we only need two points for the entire column. But it is possible, as in the last column, that after merging we might have a multi-piece column, and that for each piece we need to specify two separate points.

As shown before, to present an unsimplified shape, we needed 251 points. After simplification, however, this amount shrank to 164. But after operating the merge algorithm, these points decreased considerably and reached 12 quadrilaterals with 24 points. This points to more than one-tenth reduction in the original number of points. This considerable reduction in points from 164 to 24 had no effect of the degree of accuracy.

In the next phase, we will re-merge the bigger quadrilaterals. If the obtained quadrilaterals are the same size and are next to one another, they can merge and form a bigger one as in Fig. 10.

The bigger quadrilaterals obtained from the first phase merge, which were equal in size and situated next to one another, were merged again.

$$Qlmerg2 = \left\{ \sum_{i=i+1}^{|Qlmerg|} QLmerg_i \mid (QLmerg_{\underset{i}{Pdl}}^{x} = QLmerg_{\underset{i+1}{Pdl}}^{x}) , (\left((QLmerg_{\underset{i}{Pul}}^{x} - QLmerg_{\underset{i}{Pdl}}^{x} \right) = \left(QLmerg_{\underset{i+1}{Pul}}^{x} - QLmerg_{\underset{i+1}{Pdl}}^{x} \right)) \right\}$$

We could thereby once again reduce the number of quadrilaterals and the points required to present them.

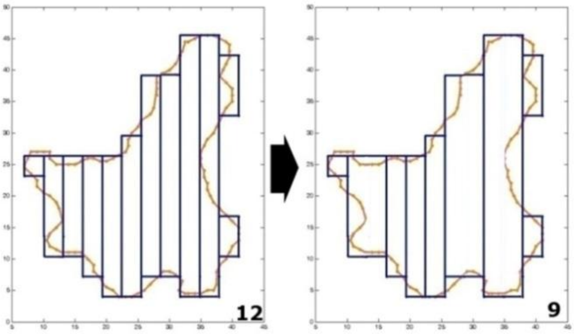

Fig. 10: Merging in the second phase in the proposed algorithm

The number of quadrilaterals decreased from 12 to 9 and required only 18 points for their presentation. It is important to note that the merging activities in the first and second phases had no impact on accuracy while the number of necessary points decreased considerably.

Apparently, the merge algorithm, which attaches quadrilaterals in columnar manner, is suitable for shapes whose heights are more than their widths. But in shapes whose widths are more than their heights, merging activity could be carried out differently. In place of merging the quadrilaterals in columnar fashion, it is possible to carry out the merging process in rows. In other words, we need to merge the quadrilaterals which are in one row and form a bigger one.

$$Qlmerg = \left\{ \sum_{i=i+ty}^{LY} QLin_i \mid QLin_{\underset{i}{Pdl}}^{y} = QLin_{\underset{i+1}{Pdl}}^{y} \right\}$$

The effect of this change could be observed in Fig. 11.The width of the shape that is employed here, as it is shown in the above figure, is more than its height. After exerting the columnar algorithm, the number of quadrilaterals reached 14. But after linear merge, this number decreased to 5, which is almost one-third of the former merge algorithm. In columnar merge we require 28 points while in linear merge only 10 points are required. Therefore, merge algorithm must take into consideration both the width and the height of the shapes, meaning that it should benefit from columnar merge in shapes whose heights are more than their widths, while it should exploit linear merge in shapes whose widths are more than their heights so that we can have the least possible number of points.

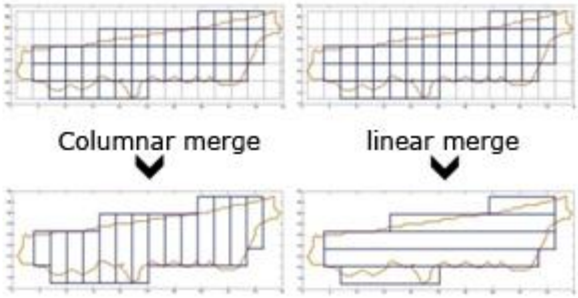

Fig. 11: Columnar and linear merge in the proposed algorithm

Our approach formulation model can be presented using the following flow chart.(Chart1)

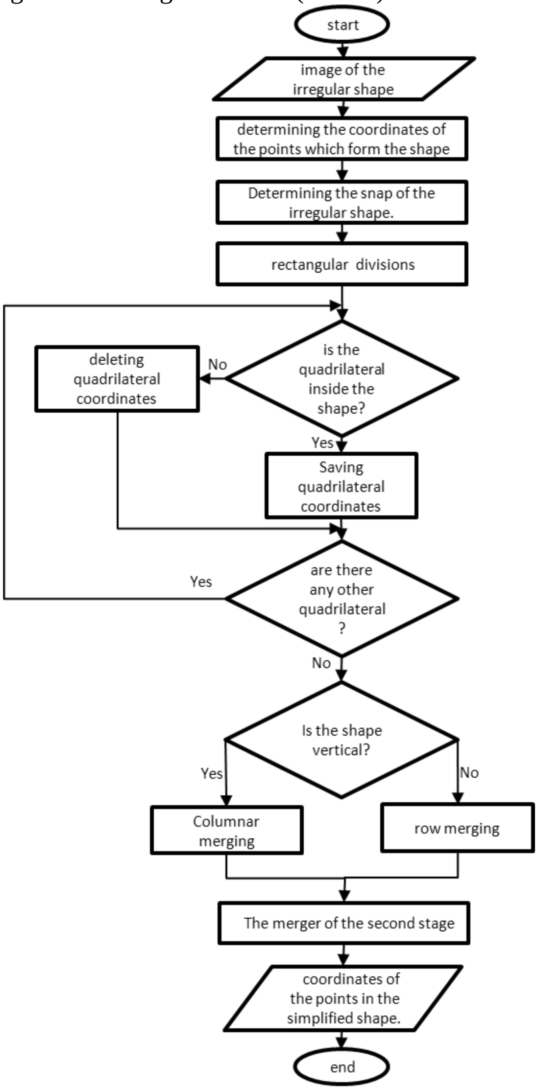

Chart. 1: Our approach formulation model

6. Implementing and Assessing the Proposed Algorithm and RDP

In this part, we have planted various sensor networks in different scenarios, simplified several different shapes with the proposed algorithm and RDP, and have sent them to the network as spatial queries. We will finally analyze the results of the queries so that it is possible for us to compare the proposed algorithm with RDP in terms of energy consumption and accuracy in responding to the queries. To carry out the plantation of sensor networks, we first need to select areas with irregular and different shapes.

We have selected three different cities as examples. We have specified the examined areas in the shapes: the first is the residential area of the city of Neyshabur (Fig. 12), next is the residential area of the city of Mashad (Fig. 13), and the third is the residential area of the city of Nur (14).

Fig. 12: Neyshabur scenario

Fig. 13: Mashad scenario

Fig. 14: Nur scenario

In this article, we implemented and tested the three scenarios using MATLAB software [29], in order to assess the obtained results.

First, we specify the constitutive points of the shape of each scenario (Fig. 15).

Fig.15:Scenarios' shapes

Then, we calculated the points and the coordinates of the shapes of the scenarios. The number of points of irregular shapes is presented in Table 1 without simplification.

Table 1: The number of points in the original shapes of scenarios

Scenario	Neyshabur	Mashad	Nur
The number of points in the original shape	251	241	207

After extracting the list of point coordinates from the scenarios, we can implement the simplification algorithm and decrease the number of points. They can now be compared with respect to the reduction of the number of points and accuracy. We first implement the RDP and then the proposed algorithm.

6.1. Implementing RDP Simplification Algorithm

The RDP algorithm was implemented on the scenarios. The number of points of irregular shapes after being simplified by RDP method in three scenarios is presented in Table 2.

Table 2: The number of points resulting from RDP algorithm

Scenario	Neyshabur	Mashad	Nur
The number of points of unsimplified shape	251	241	207
The number of points of the shape simplified with RDP	99	102	94

Using RDP algorithm, we could reduce the number of points to more than half the original number. However, after planting the sensor network, it has to be compared with the proposed algorithm in terms of accuracy and the number of points.

6.2. Implementing the Proposed Simplification Algorithm

In this section, we implemented the proposed algorithm and the merge function on the three scenarios.

First, we carried out quadrilateral divisions according to what was previously mentioned, and presented the results in Fig. 16. The coordinates of each corner of the quadrilateral were recorded in a matrix. In Table 3, we can observe the number of quadrilaterals formed for each scenario.

Table 3: The number of quadrilaterals resulting from rectangular formations in the scenarios

Scenario	Neyshabur	Mashad	Nur
Number of quadrilaterals	1394 (41×34)	1610 (35×46)	644 (14×46)

Now, we should see which quadrilaterals fall within the shapes. The results obtained from the examination are presented in the following. As presented in Fig. 16, the quadrilaterals whose two corners are inside the shape are considered as being entirely inside, and their sides are highlighted. The number of quadrilaterals together with the number of the ones which are considered as inside the shape are presented in Table 4.

According to Table 4 and Fig. 16, the number of the quadrilaterals is relatively large. Also, many of these quadrilaterals are close to one another and are capable of forming a bigger quadrilateral.

In the following, we will implement the merge algorithm of the first phase on these quadrilaterals. The results obtained from the operation of merging the quadrilaterals are presented in Fig.16. In the second phase, bigger and equal quadrilaterals obtained from the merge algorithm of the first phase can be further merged. The merge algorithm of the second phase was implemented, and its results are presented in Fig. 16.

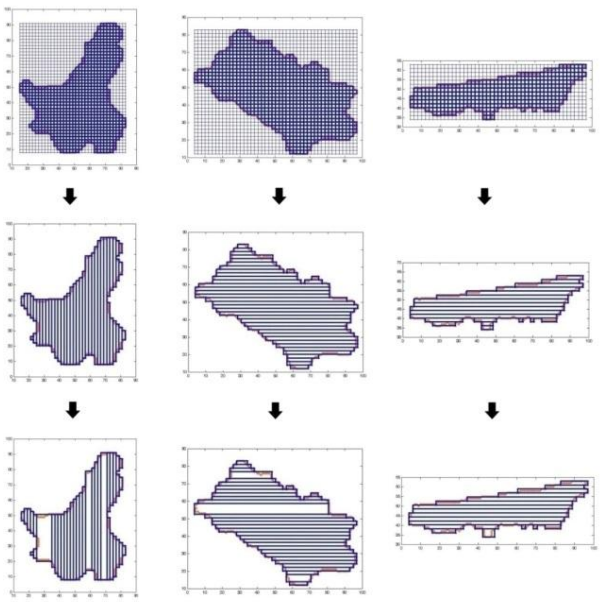

Fig. 16: The implementation of two merges phases on the scenario using the proposed algorithm.

According to Fig. 16, by merging quadrilaterals, we were able to considerably decrease their number. The number of quadrilaterals is presented in Table 4.

Table 4: The number of quadrilaterals obtained from implementing the proposed algorithm in different phases.

Scenario	Neyshabur	Mashad	Nur
Total number of quadrilaterals	1394	1610	644
Number of quadrilaterals inside the shape	758	830	355
Number of quadrilaterals after the first phase merge	43	38	19
Number of quadrilaterals after the second phase merge	37	34	18

It should be noted that scenarios are merged with respect to their width and height. For instance, Neyshabur scenario, whose height was more than its width, was merged in a columnar fashion. On the other hand, Mashad and Nur scenarios were merged in rows and horizontally, since their widths were more than their heights. Since for the presentation of each quadrilateral we require two points, the ultimate required points for the presentation of the scenarios with different algorithms are presented in Table 5.

Table 5: The number of points obtained from simplification using RDP and the proposed algorithms.

Scenario	Neyshabur	Mashad	Nur
Unsimplified number of points	251	241	207
Number of points obtained from RDP algorithm	99	102	94
Number of points obtained from the proposed algorithm	74	68	36

6.3. Calculating Carelessness

The simplification of the shape of the areas and the reduction in the number of points to present these areas caused a certain degree of carelessness. There is a difference between the shapes simplified with simplification algorithm and the original shapes. This is because after simplification, some parts which formerly fell inside the shape, would now fall outside it, while some parts which did not belong to the shape, were now annexed to it. These differences give rise to a certain degree of carelessness in responses to the queries because some of the nodes which in the original shape were inside the area now fall outside it. And some of the nodes which were outside the original shape are now inside it. These differences exist normally around the shape of the intended area. This carelessness explains the reason why some nodes that should not have responded to the queries transmit some responses, while the nodes that ought to have responded to the queries do not participate in sensing process. This degree of carelessness is calculated by the following formula.

Carelessness= the proportion of the number of wrong nodes to the total number of nodes in the area in the original shape.

The number of wrong nodes is equal to the total number of nodes that wrongly responded + the nodes that did not respond wrongly.

6.4. The Implementation of Sensor Network in the Scenarios

Now, we deploy the sensor network in three different scenarios so that we can compare them with respect to the maximum level of energy consumption and accuracy in different scenarios and in the simplification algorithm. To

make this comparison, we need to deploy the sensor network in the original shape with regard to which the two simplification algorithms can be compared.
-The dimensions of the deployment area are 100*100 m.
-The queries are broadcast through fullflooding method.
- Only the nodes inside the area respond to queries.
-Network packets are 28 bytes.
- Nodes are aware of their position (they are equipped with GPS).
-The energy of data transmission is 50nj/bit.
- We require two bytes of memory to send each point of the shape.
- Nodes are identical and stable.
- The radio range of each node is 15m.
- Nodes are always active and never change into sleep mode.
-Each node is identified from 1 to *n* with a unique identity information (n is the number of nodes of the network).
-The consumed energy is equal for both sending and receiving data.
Each scenario ought to be deployed with different number of nodes (500-1000-1500). To show the degree of carelessness, we benefit from presenting the percentage of wrong nodes to the total number of nodes of the area in the original shape. This means that we first should obtain the number of wrong nodes, and then express them as the percentage of the total number of nodes in the original shape.

6.5. The Implementation of Sensor Network in Neyshabur Scenario

First, we implement the sensor network in Neyshabur scenario with the original shape, simplified with RDP algorithm, and, finally, with the proposed algorithm. The results obtained from the implementation of the sensor network with different algorithm are available in Table 6. Based on Table 6, the level of energy consumption of the sensor network in Neyshabur scenario with RDP and our proposed algorithms, and the implementation without the simplification of the points (main shape) with different number of nodes (500-1000-1500) are presented in Fig. 17.

Table 6: The results of the sensor network implementation in Neyshabur scenario

Simplification by proposed algorithm			Simplification by RDP algorithm			Main Shape			Neyshabur scenario Sink coordinates 29* 83
1500	1000	500	1500	1000	500	1500	1000	500	Number of nodes
480	313	152	522	338	161	469	310	149	Nodes inside the shape
74	74	74	99	99	99	251	251	251	Number of points
6	6	6	7	7	7	18	18	18	Number of packets
9.3	4.1	1.0	10.9	4.8	1.2	28.1	12.5	3.2	Energy consumption (Jules)
7	6	0	4	3	2	Nodes that should have responded			
18	9	3	55	34	14	Nodes that should not have responded			
5.3	4.8	2	12.5	11.9	10.7	Error percentage			
66.6	66.6	66.5	61.6	61.0	60.9	Percentage of energy consumption reduction			

By decreasing the number of points of the original shape in Neyshabur scenario, and also by reducing network packets, the simplification algorithms could considerably decrease energy consumption. This reduction in energy consumption occurred in all three networks with different number of nodes (500-1000-1500).

As it is obvious from Fig. 17, our proposed algorithm outperformed RDP algorithm and could respond to the queries by consuming less energy. The amount of reduction in energy consumption in Neyshabur scenario for 500-1000-1500 number of nodes in RDP algorithm was 60.99, 61.04, and 61.06, respectively.
However, the energy reduction in our algorithm was 66.55, 66.60, and 66.62, respectively: 5% more reduction in energy consumption relative to other algorithms. This is due to the better performance of the proposed algorithm in reducing the number of points in the shape.

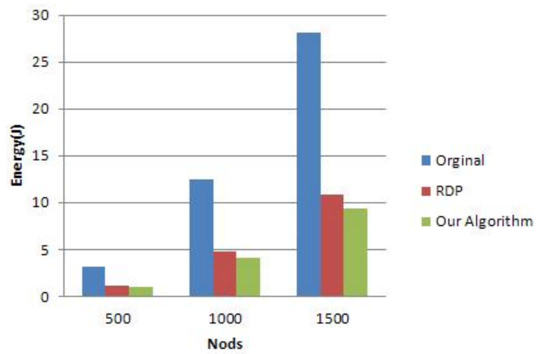

Fig. 17: Comparing algorithms in Neyshabur scenario with respect to energy consumption

The carelessness percentage of Neyshabur scenario is available in Fig. 18. This diagram illustrates the carelessness percentage in two RDP and the proposed algorithms in relation to the unsimplified original shape. In Neyshabur scenario, our proposed algorithm enjoyed lower carelessness percentage in relation to RDP algorithm. This degree of lower carelessness occurred in each mode of implementation of sensor network.

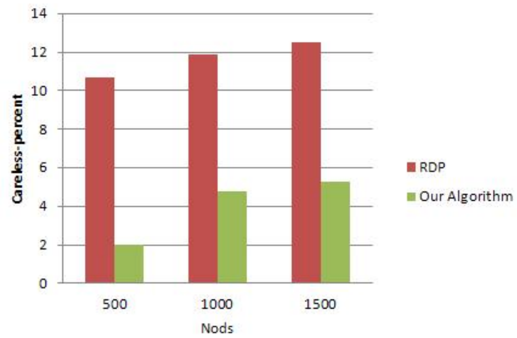

Fig. 18: Comparing algorithms in Neyshabur scenario with respect to carelessness

The RDP algorithm in the sensor network with different nodes (500-1000-1500) suffered from carelessness 10.7, 11.9, 12.5 respectively. In our proposed algorithm, carelessness was 2, 4.8, and 5.3 respectively. Consequently, in terms of accuracy, our proposed algorithm outperformed RDP algorithm. The lower carelessness is due to a more accurate simplification of our algorithm since fewer wrong areas were added and subtracted from the shape. These wrong areas will be explicated in section 6-3.

The carelessness of our algorithm in Neyshabur scenario was, in the worst possible state, was 5.3 per cent, while the carelessness of RDP algorithm, in the worst possible state, reached 12.5 per cent.

6.6. The Implementation of Sensor Network in Mashad Scenario

In this section, we first implement the sensor network in Mashad scenario with the original shape. Next, we will implement the simplified sensor network using RDP algorithm. And, finally, using our proposed algorithm, we will deploy it in the same scenario.

The results obtained from the implementation of the sensor networks with different algorithms are presented in Table 7.

Figure 19 presents the results of the implementation of Mashad scenario with regard to energy consumption. This implementation of the sensor network is achieved with 500-1000-1500 nodes. The following diagram illustrates the implementation of RDP simplification and our proposed algorithms on the original shape of Mashad scenario.

In this scenario, too, our algorithm outperformed RDP algorithm, while at same time, by a better simplification of the points, it could lessen energy consumed in responding to the queries. This reduction was even better than Neyshabur scenario, and as the results reveal, it could, in all the three sensor networks with different node numbers, have lesser energy consumption. The results reveal that reduction in energy consumption in RDP algorithm in sensor networks with 500-1000-1500 number of nodes was 52.78, 52.85, and 53.21 per cent respectively, while in our proposed algorithm this reduction for the same number of nodes was 70.43, 70.50, and 70.73 per cent respectively.

Table 7: Results of the implementation of sensor network in Mashad scenario

Simplification by the proposed algorithm			**Simplification by RDP algorithm**			**Original shape**			**Mashad scenario** Sink coordinates 48* 97
1500	1000	500	1500	1000	500	1500	1000	500	**Number of nodes**
480	313	152	573	391	184	514	345	158	**Nodes inside**
74	74	74	102	102	102	241	241	241	**Shape's number of nodes**
6	6	6	8	8	8	17	17	17	**Number of packerts**
9.3	4.1	1.0	12.5	5.5	1.4	26.7	11.8	3.0	**Energy consumption (Jules)**
7	6	0	2	1	0	**Nodes that should have responded**			
18	9	3	59	46	26	**Nodes that should not have responded**			
5.3	4.8	2	11.8	13.6	13.9	**Error percentage**			
66.6	66.6	66.5	52.8	52.8	52.7	**Energy consumption reduction**			

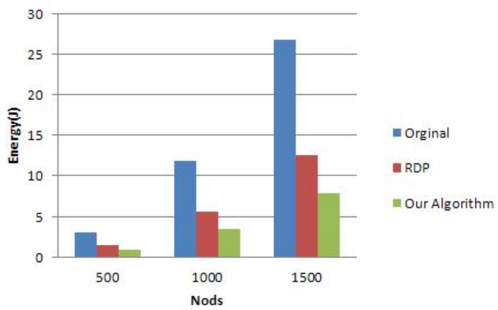

Fig.19:Comparing the algorithms in Mashad scenario in terms of energy consumption

Accordingly, our proposed algorithm could considerably optimise energy consumption by more than 17%. This reduction is due to the better performance of the proposed algorithm in reducing the number of points of the shape. The results also reveal that our proposed algorithm had a finer performance in Mashad scenario than in Neyshabur scenario, and reduction in energy consumption could reach from 5% to 17%. Fig. 20 illustrates the degree of carelessness in Mashad scenario. The degree of carelessness is calculated in percentage terms after the implementation of the sensor network with different number of nodes and the deployment of the algorithms.

Our algorithm in Mashad scenario had far lesser carelessness, such that it could even be ignored. This lesser degree of carelessness is down to a more accurate simplification of our algorithm because fewer wrong areas were added or subtracted from it. These wrong areas were explained in section 6-3. In the worst possible condition, RDP algorithm suffered from 14% error, while the error of our algorithm was, in the worst possible condition, only 4%.

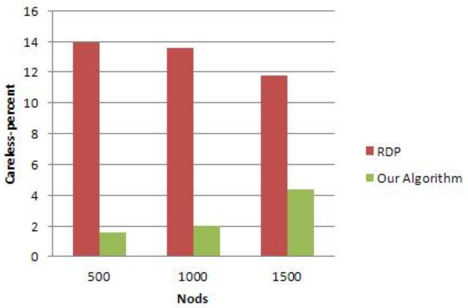

Fig.20:Comparing the algorithms in Mashad scenario in terms of carelessness

After deploying the sensor network with 500-1000-1500 nodes, the degree of carelessness of RDP in Mashad scenario was 13.9, 13.6, and 11.8 per cent respectively. This parameter for our proposed algorithm with the same number of nodes was 1.6, 2, and 4.4 per cent respectively. The error in the sensor network with 500 nodes was close to zero.

6.6.The Implementation of Sensor Network in Nur Scenario

In the following, we will implement the senor network in Nur scenario with the original shape, and after that with the shape simplified with RDP algorithm, and finally with our proposed algorithm. The results the implementation of the sensor network with different algorithms are presented in Table 8.

Table 8: Results of the implementation of sensor network in Nur scenario

Simplification by the proposed algorithm			Simplification by RDP algorithm			Original Shape			**Nur scenario** Sink coordinates 58* 97
1500	1000	500	1500	1000	500	1500	1000	500	Number of nodes
223	136	67	242	154	76	235	139	71	Nodes inside
36	36	36	94	94	94	207	207	207	Shape's number of points
3	3	3	7	7	7	15	15	15	Number of packets
4.6	2.0	0.5	10.9	4.80	1.2	23.4	10.4	2.6	Energy consumption (Jules)
13	6	5	8	3	1	Nodes that should have responded			
1	3	1	7	15	5	Nodes that should not have responded			
5.9	6.4	8.4	6.3	12.9	8.4	Error percentage			
79.9	79.9	79.9	53.3	53.2	53.2	Energy reduction percentage			

In Fig. 21 the results of the energy consumption of this implementation with RDP and our proposed algorithms are presented against the implementation on the original shape of Nur scenario.

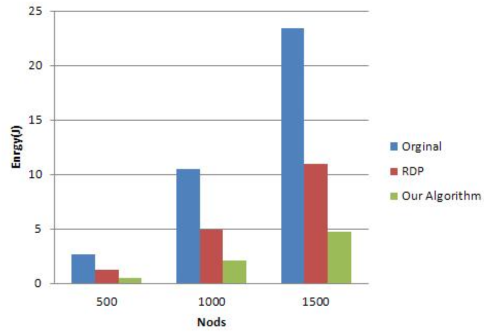

Fig. 21: Comparing the algorithms in Nur scenario in terms of energy consumption

As Fig. 21 shows, a better result was obtained by our algorithm. The results show that the sensor network with different node numbers implemented with our algorithm in Nur scenario had a far lesser energy consumption in relation to RDP algorithm. Based on implementation results, reduction in energy consumption in Nur scenario was even better than previous scenarios. Considering the results of the deployment of the sensor network with different number of nodes in Nur scenario, RDP algorithm with 500-1000-1500 nodes could enjoy 53.25, 53.29, and 53.30 per cent reduction in energy consumption respectively.

However, our algorithm has considerably decreased energy consumption by 79.92, 79.95, and 79.96 per cent respectively. This reduction in energy consumption is due to the better performance of the proposed algorithm in reducing the number of points in the shape. Energy reduction in Nur scenario was more than those of Neyshabur and Mashad, and our algorithm could reduce energy consumption by 80% in Nur scenario. It could respond to the queries consuming half the amount of the energy consumed in RDP algorithm. The results of carelessness obtained from the deployment of the sensor network in Nur scenario with different number of nodes are presented in Fig. 22.

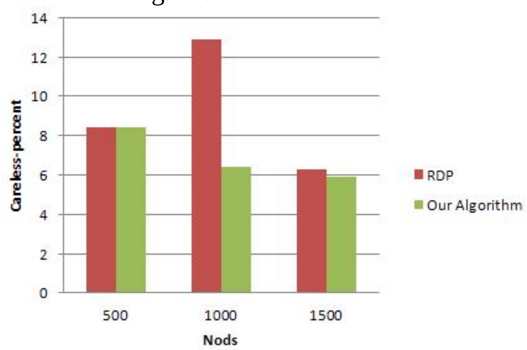

Fig.22:Comparing the algorithms in Nur scenario in terms of carelessness

In this scenario, the two RDP and proposed algorithms had similar performance with regard to carelessness, except when the number of nodes was 1000. The RDP algorithm in the sensor network with 500-1000-1500 nodes achieved 8.4, 12.9, and 6.3 per cent carelessness respectively. The proposed algorithm, too, achieved 8.4, 6.4, and 5.9 per cent carelessness respectively. This close degree of carelessness is due to the formation of equal wrong areas by both of the algorithms. These wrong areas were explained in section 6-3. The consumed energy by the proposed algorithm in this scenario was half the amount consumed by RDP algorithm. But, as the diagram reveals, carelessness in both algorithms is more or less equal. it could be argued that as long as carelessness in both algorithms is equal, the energy consumed by our algorithm is half the RDP algorithm.

7. Conclusion

In this article we introduced the processing of spatial queries in sensor networks and analysed its function in crisis management and stricken areas. The problem with these areas was their irregular shape which required many points to be presented. This resulted in an increase in the sensor network's number of packets and higher energy consumption. By examining previous related works and benefiting from line simplification algorithms, we could reduce the points of the shape of the area and solve the problem. Then, we presented our algorithm and, as the results of implementation revealed, in all the scenarios the proposed algorithm manifested a better performance in reducing energy consumption. In the worst-case scenario, our proposed algorithm reduced energy consumption by 66%, while in the best-case scenario, it reduced energy consumption by 80% relative to the original shape of the area. This amount of reduction of energy consumption in the sensor network which has limited energy is a notable success. In order to draw a satisfactory conclusion from the results{Vieira, 2003 #1} of the implementations and include all the scenarios with their different shapes, we worked out an average of energy reduction for both the proposed and RDP algorithm which is presented in Fig. 23. According to this Figure, on average, RDP and our proposed algorithms reduced energy consumption by 55.7% and 72.3% respectively. This amount of energy reduction was due to the better performance of the proposed algorithm in reducing the number of points of the original shape. Our proposed algorithm, therefore, performed almost 17% better than the last solution in tackling the problem of irregular shapes.

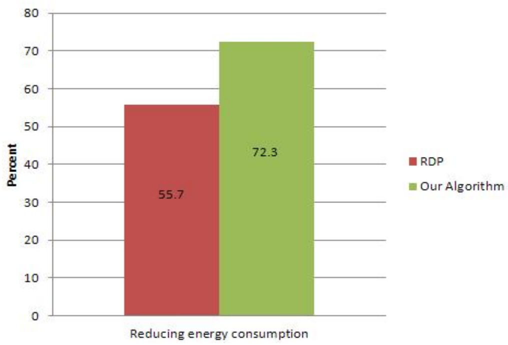

Fig. 23: Comparing RDP and the proposed algorithm in all the scenarios in terms of reducing energy consumption

We calculated the average of carelessness parameter in the implementations. As Fig. 24 shows, RDP algorithm responded to the queries with on average 11.3% carelessness. This level of carelessness in the previous works which used RDP algorithm remained the same.

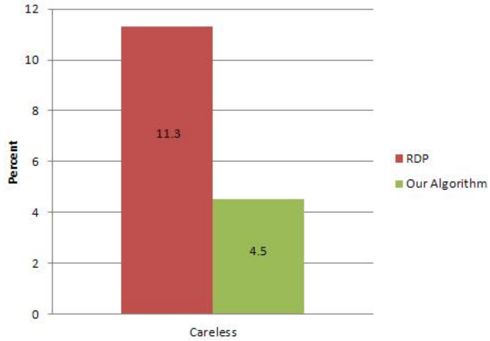

Fig. 24: comparing RDP and the proposed algorithm in the scenarios in terms of carelessness percentage.

However, our proposed algorithm could respond to queries with 4.5% carelessness. This lesser degree of carelessness was due to a more accurate simplification by our algorithm since fewer wrong areas were added and subtracted from the shape. This 4.5% degree of carelessness against 72.3% energy reduction can be ignored.

The most important achievement of this research is presenting an spatial query mechanism in sensor network that, when confronted with irregular areas, can have the best performance and, with the lowest degree of energy consumption, respond to queries. With respect to lesser energy consumption and maximised network lifetime, it can respond to more queries so that more human lives could be saved at times of crisis.

In future researches we can improve the quadrilateral merging phase because in this research we merged the quadrilaterals in columns and in horizontal rows. Since the number of quadrilaterals is high, there are numerous ways to merge them, while only the best one ought to be selected so that we can experience the highest reduction if the number of points. Using genetic algorithm is recommended. The next stage which has the potential to be improved is the act of determining whether the quadrilaterals are inside or outside the shape. If two angles of the quadrilateral are inside the shape, we consider it as inside, which is not very accurate though. If we can decide on the part of the quadrilateral which falls inside the shape, carelessness decreases.

References

[1] J. Löffler, J. Schon, V. Hernandez-Ernst, J. Pottebaum, and R. Koch, "Intelligent use of geospatial information for emergency operation management," in Proceedings of the fourth international conference on information systems for crisis management, ISCRAM, 2007.

[2] A. Soheili, V. Kalogeraki, and D. Gunopulos, "Spatial queries in sensor networks," in Proceedings of the 13th annual ACM international workshop on Geographic information systems, 2005, pp. 61-70.

[3] M. Careem, C. De Silva, R. De Silva, L. Raschid, and S. Weerawarana, "Demonstration of Sahana: free and open source disaster management," in Proceedings of the 8th annual international conference on Digital government research: bridging disciplines & domains, 2007, pp. 266-267.

[4] P. K. Chitumalla, D. Harris, B. Thuraisingham, and L. Khan, "Emergency response applications: dynamic plume modeling and real-time routing," IEEE Internet Computing, vol. 12, pp. 38-44, 2008.

[5] M. A. M. Vieira, C. N. Coelho, D. da Silva, and J. M. da Mata, "Survey on wireless sensor network devices," in Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference, 2003, pp. 537-544.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Computer Networks, vol. 52, pp. 2292-2330, 2008.

[7] J. Zheng and A. Jamalipour, Wireless sensor networks: a networking perspective: John Wiley & Sons, 2009.

[8] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," ACM Transactions on database systems (TODS), vol. 30, pp. 122-173, 2005.

[9] Z. Can and M. Demirbas, "A survey on in-network querying and tracking services for wireless sensor networks," Ad Hoc Networks, vol. 11, pp. 596-610, 2013.

[10]J. Gehrke and S. Madden, "Query processing in sensor networks," IEEE Pervasive Computing, vol. 3, pp. 46-55, 2004.

[11]A. Coman, M. A. Nascimento, and J. Sander, "A framework for spatio-temporal query processing over wireless sensor networks," in Proceeedings of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB 2004, 2004, pp. 104-110.

[12]Google map - a desktop web mapping service. Available: https://maps.google.com

[13]W. R. Tobler, "AN EXPERIMENT IN THE COMPUTER GENERALIZATION OF MAPS," DTIC Document1964.

[14]A. H. Robinson, J. Morrison, P. C. Muehrcke, A. Kimerling, and S. Guptill, "Elements of cartography, john wiley&sons," Inc., New York, USA, 1995.

[15]P. S. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," DTIC Document1997.

[16]D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, pp. 112-122, 1973.

[17]U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," Computer graphics and image processing, vol. 1, pp. 244-256, 1972.

[18]T. Gökgöz, A. Sen, A. Memduhoglu, and M. Hacar, "A New Algorithm for Cartographic Simplification of Streams and Lakes Using Deviation Angles and Error Bands," ISPRS International Journal of Geo-Information, vol. 4, pp. 2185-2204, 2015.

[19]P. Chandrasekara, T. Mahaulpatha, D. Thathsara, I. Koswatta, and N. Fernando, "Landmarks based route planning and linear path generation for mobile navigation applications," Spatial Information Research, pp. 1-11, 2016.

[20]M. Lucena, J. M. Fuertes, A. L. Martinez-Carrillo, A. Ruiz, and F. Carrascosa, "Efficient classification of Iberian ceramics using simplified curves," Journal of Cultural Heritage, vol. 19, pp. 538-543, 2016.

[21]A. Jelinek, L. Zalud, and T. Jilek, "Fast total least squares vectorization," Journal of Real-Time Image Processing, pp. 1-17, 2016.

[22]R. I. da Silva, D. F. Macedo, and J. M. S. Nogueira, "Duty cycle aware spatial query processing in wireless sensor networks," Computer Communications, vol. 36, pp. 149-161, 2013.

[23]W. Li, Y. Li, P. Yu, J. Gong, and S. Shen, "The Trace Model: A model for simulation of the tracing process during evacuations in complex route environments," Simulation Modelling Practice and Theory, vol. 60, pp. 108-121, 2016.

[24]R. I. da Silva, V. D. D. Almeida, A. M. Poersch, and J. M. S. Nogueira, "Spatial query processing in wireless sensor network for disaster management," in 2009 2nd IFIP Wireless Days (WD), 2009, pp. 1-5.

[25]N. Fernández-García, L. D.-M. Martínez, A. Carmona-Poyato, F. Madrid-Cuevas, and R. Medina-Carnicer, "A new thresholding approach for automatic generation of polygonal approximations," Journal of Visual Communication and Image Representation, vol. 35, pp. 155-168, 2016.

[26]R. I. Da Silva, D. F. Macedo, and J. M. S. Nogueira, "Spatial query processing in wireless sensor networks–A survey," Information Fusion, vol. 15, pp. 32-43, 2014.

[27]W.Cao, Y. Li, "DOTS: An online and near-optimal trajectory simplification algorithm",Systems and Software, Vol 126, Pages 34–44, 2017.

[28] V. Canellidis, J. Giannatsis, V. Dedoussis "Evolutionary Computing and Genetic Algorithms: Paradigm Applications in 3D Printing Process Optimization", Intelligent Computing Systems
Vol 627 , pp 271-298, 2016.

[29]"MATLAB - Simulink-Mathematical Software."

Mohammad Shakeri received the B.Sc. degree in Islamic Azad university of Neyshabur, Neyshabur, Iran, in 2011. He received the M.Sc. degree in software engineering from Khorasan Razavi,Neyshabur, Science and Research branch, Islamic Azad university, Neyshabur, Iran, in 2016. He is currently work in Department of Information Technology in blood transfusion organization of Khorasan Razavi . His area research interests include Computer Networks, Wireless Sensor Networks and Image Processing. His email address is: Alborz.corp@gmail.com

Sayyed Majid Mazinani was born in Mashhad, Iran on 28 january 1971. He received his Bachelor degree in Electronics from Ferdowsi University , Mashhad, Iran in 1994 and his Master degree in Remote Sensing and Image Processing from Tarbiat Modarres University, Tehran, Iran in 1997. He worked in IRIB from 1999 to 2004. He also received his phD in Wireless Sensor Networks from Ferdowsi University , Mashhad, Iran in 2009. He is currently assistant professor at the faculty of Engineering in Imam Reza International University, Mashhad, Iran. He was the head of Department of Electrical and Computer Engineering from 2009 to 2012. His research interests include Computer Networks, Wireless Sensor Networks and Smart Grids.