Disclaimer

The University of Gloucestershire has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

The University of Gloucestershire makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

The University of Gloucestershire makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

The University of Gloucestershire accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 9169236, 9 pages
http://dx.doi.org/10.1155/2016/9169236

Research Article

Coverage Enhancement Algorithms for Distributed Mobile Sensors Deployment in Wireless Sensor Networks

Muhammad Sirajo Aliyu,[1] Abdul Hanan Abdullah,[1] Hassan Chizari,[1] Thabit Sabbah,[1] and Ayman Altameem[2]

[1] *Faculty of Computing, Department of Computer Science, Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia*
[2] *College of Applied Studies and Community Services, King Saud University (KSU), Riyadh, Saudi Arabia*

Correspondence should be addressed to Muhammad Sirajo Aliyu; muhammadsirajo@yahoo.com

Received 22 July 2015; Revised 12 February 2016; Accepted 15 February 2016

Academic Editor: Jaime Lloret

Sensor nodes in wireless sensor networks are deployed to observe the surroundings for some phenomenon of interest. The fundamental issue in observing such environments is the area coverage which reflects how well the region is monitored. The nonuniform sensor nodes distribution in a certain region caused by random deployment might lead to coverage holes/gaps in the network. One of the solutions to improve area coverage after initial deployment is by sensor nodes mobility. However, the main challenge in this approach is how to increase area coverage with the least energy consumption. This research work aims to improve area coverage with minimal energy consumption and faster convergence rate. The Edge Based Centroid (EBC) algorithm is presented to improve the area coverage with faster convergence rate in a distributed network. The simulation based performance evaluations of the proposed algorithms are carried out in terms of area coverage, convergence rate, and energy efficiency. Compared to the existing works, EBC improved area coverage with faster convergence. It is concluded that the proposed algorithm has improved area coverage with faster convergence and minimal energy consumption.

1. Introduction

Wireless sensor networks (WSNs) have gained considerable worldwide attention in different research communities in recent years, due to their widespread applications in many scenarios. Such applications include battlefield surveillance, forest fire detection, disaster monitoring, and postdisaster operations [1–4]. They significantly improve the capabilities to control and monitor the physical environment. The utilization of sensor networks is transforming the traditional techniques of collecting data, linking the gap between physical and virtual world [5, 6]. In WSNs, sensor nodes can be deployed randomly or deterministically; however, they should be appropriately placed to achieve sufficient area coverage level for better monitoring of sensing field [7–9].

Although the deterministic deployment is applicable in some application, there are some working environments that are related to natural disaster areas, harsh environments, and toxic regions in which sensor deployment cannot be deterministically or manually deployed. Deploying sensor nodes with the aid of an aircraft or similar vehicles randomly is one of the possible ways in such environments. However, using this technique (random deployment), the initial landing positions of sensors cannot be uniform or controlled, due to the existence of obstacles such as buildings, trees, and wind, causing some areas of the sensing field (SF) to be more dense than others [10]. Thus, the sensor nodes may not adequately cover the SF, and the highly dense area will cause the sensor network to be an unusual or abnormal network [11, 12]. Therefore, it is essential to make use of mobile sensors, which can move iteratively to a better location that can give the required coverage.

The main goal of mobile sensor network is to improve the area coverage [13]. To achieve this goal, a group of sensors have to be relocated and deployed in more suitable locations in order to monitor the SF effectively. Mainly, WSNs in terms of mobility control are of two types: centralized and distributed. In centralized WSNs, sensors are controlled by a sink node or a central server. However, in the distributed networks, sensors are self-controlled.

Due to the nature of distributed networks that is more appealing to minimize the information exchange between the sensor nodes, no sensor has a priori information of the SF and initial positions of other sensors [14]. In addition, each sensor node has limited sensing and communication abilities which make the sensor nodes unable to obtain the entire network information [15, 16]. Therefore, sensors are deployed randomly and allowed to move and communicate with their neighbours by exchanging information related to each of them. The advantage of this kind of deployment is that sensors can be deployed and work in environments that are harsh or not easily accessible. Commonly, the SF is partitioned into small convex polygons named Voronoi [17–19], such that each polygon will contain only a single sensor node that only needs to cover its own polygon.

In this paper, a distributed movement assisted sensor deployment algorithm, the Edge Based Centroid (EBC), is proposed to enhance area coverage in a distributed mobile sensor network which can be valuable in terms of area coverage. The EBC algorithm tends to move sensor nodes towards the centroid of local Voronoi polygon, from their initial deployment position. The sensor nodes movement is iterative; this is one of the proposed algorithm's main features, in addition to the certainty of coverage increment or stability after every iteration.

The rest of this paper is organized as follows. Section 2 presents some of the related works. Section 3 describes the proposed distributed algorithm of mobile sensors. Section 4 presents the proposed EBC method. Section 5 shows the simulation results and discussion. Lastly, Section 6 is the conclusion of the paper.

2. Related Works

Recently, there exist significant works on sensor deployment and coverage related issues which have kept the area as an active research domain. Directed coverage is used to monitor the area in between two boundaries by a sensor network [20]. The appropriate location to place sensors nodes is vital [21], in monitoring smaller or larger region [22]. Sufficient coverage is essential to WSN [23], while the improvement of coverage is important after the initial deployment of sensor nodes. The distributed movement strategy is commonly used to improve the area coverage of MF by moving the sensor nodes from one location to another [24, 25]. The initial locations of deployed sensor nodes, and where to relocate them, are vital to network coverage [14]. In most of the distributed movement algorithms, monitoring field (MF) is partitioned into subareas, with different sensor node assigned to a particular location. These partitioned subareas cover the entire MF; if the sensor node assigned to monitor a particular subarea cannot detect any expected event within its subarea, no other sensor node can detect it [26]. Therefore, examination of each subarea for coverage hole and calculation of new candidate location is the responsibility of deployed sensor node in that subarea. When a sensor node is unable to cover its own subarea, then there exists a coverage hole (uncovered area).

The challenge before many current distributed movement assisted algorithms is the calculation of a new candidate location within its subarea which is not considered before relocating sensor node [27]. This has constituted a main concern to researchers for coming up with solutions that will address such problem in mobile wireless sensor network [24, 28]. In spite of calculating the new candidate location in its subarea that will enhance area coverage, many of the earlier and some current works proposed algorithms that were unable to cover their subarea adequately. However, a network with coverage holes will not be able to satisfactorily carry out its monitoring task. On the other hand, most of the distributed movement algorithms were designed to have much iteration in their local region before determining the new position. At the end of iterations, this leads to higher energy consumption incurred by sensor nodes movements.

To address those issues of sensor node relocation to a new candidate location that will enhance the area coverage, various approaches have been proposed [28, 29]. Wang et al. (2006) proposed three different distributed movement assisted sensor deployment algorithms, VEC, VOR, and Minimax, to improve the total area coverage. Voronoi diagram is used to partition the MF into n convex polygons where every polygon enclosed one sensor node only. The methods use their local polygon information to calculate the new candidate location to move sensor node. In comparison with other techniques, the VEC approach uses virtual force between two sensor nodes to push them away from each other at a certain distance. Minimax and VOR algorithms are greedy, which try to fix the largest coverage hole by moving sensor node towards the farthest polygon vertex. The sensor node close to a narrow edge of its polygon did not require it to move towards the farthest vertex. The result of such movement may not reduce coverage hole, but rather it can increased in some cases. The two vertex base methods, Minimax and VOR algorithms, are designed to move towards the farthest Voronoi polygon vertex to reduce coverage hole. But in some cases moving towards the farthest vertex with the aim of decreasing the distance may not always reduce the coverage hole, because some polygons may have a narrow shape. To further address the stated problem, four local movement methods are proposed in [24]. The calculation of new candidate locations in those algorithms is by considering circles whose centre is positioned within their respective polygons. There are some cases (rounds) in which the centroid of those circles is outside the polygons, meaning that there is no movement in that round by the affected sensor nodes. This problem makes the algorithms have more rounds before the algorithm converges. The more the rounds it takes, the more the sensor resources are being consumed; examples of resources include energy since sensors have to communicate before constructing their local polygon. Therefore, the proposed EBC distributed movement assisted algorithm addresses the highlighted problems in existing works discussed, which serves as the main differences between the proposed and current work.

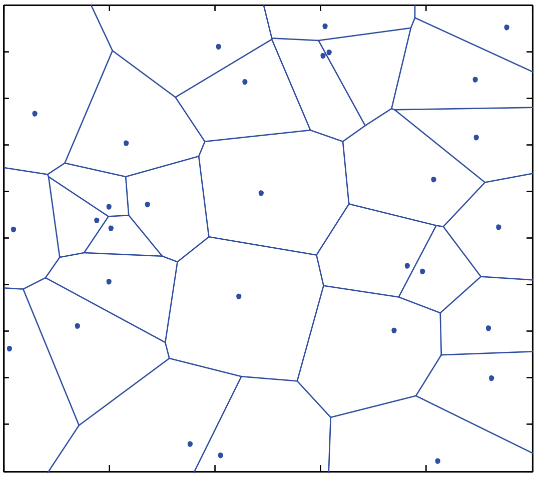

Figure 1: Voronoi diagram.

3. Preliminaries

3.1. Sensing Models. The sensing model describes the possibility of event detection by sensor in monitoring field [30]. Every sensor type has its own unique sensing model which is characterized by its sensing area, accuracy, and resolution. The sensing area relies on numerous factors such as the signal strength generated from the source, distance between the sensor node and source, the rate of attenuation in propagation, and the required self-confidence sensing level [30]. In [31], a network of acoustic sensors is deployed to detect mobile vehicles. The sensor nearer to a vehicle can detect higher acoustic signal strength than the one farther away from the vehicle due to signal attenuation, and as a result there is higher confidence of detecting vehicles [31]. In this study, isotropic sensing model is considered, which means identical sensing ability in all directions [32]. Each sensing area is associated with a sensor node that is represented by a circle with the same radius. When comparing sensing coverage algorithms, this is a common assumption as in [7, 33, 34].

3.2. Voronoi Diagram. The Voronoi diagram is a basic method of partitioning an area [19]. It is one of the important computational geometry tools for resolving coverage problem of WSNs [35, 36]. The Voronoi diagram of a collection of points partitions a given place into several polygons, which represents the proximity information about a set of geometric nodes; Figure 1 is an example of Voronoi diagram. Every point in a given polygon is closer to the node in that polygon than to any other node in the plane, which is known as generating point [37]. Let us consider a given set of network sensors by $S = \{s_1, s_2, s_3, \dots, s_n\}$ in the two-dimensional plane; then, the area can be partitioned into N convex polygons such that each sensor is the generating point of that polygon which is known as Voronoi diagram as shown in Figure 1.

4. Edge Based Centroid Algorithm

In this section, detailed description of EBC algorithm is given; it includes how to carry out the collection of location information, calculation of new candidate location, sensor node relocation, and algorithm termination. The proposed method is based on Voronoi diagram that represents the nearest information about a set of sensors. Sensing and detection of an expected event in a local polygon are the responsibility of each sensor. The sensor nodes use iterative movement for relocation from one location to another in this algorithm. This is the main feature used by this method until the termination condition is fully satisfied.

4.1. Characteristics of the Network. Before going into the method of sensor deployment, it is important to highlight different assumptions of the sensor networks. The following network features are assumed:

(i) The sensing field is a flat surface without any obstacle.

(ii) All the sensor nodes are homogeneous and autonomous with the ability to locate themselves in the sensing field.

Communication topology is connected as in [38], so that sensor location information about other sensors can be obtained through appropriate communication routes. This will enable accurate calculation of its polygon.

4.2. Location Information Collection. In the first place, every sensor node (S_i, $i \in n$) deployed will begin to broadcast its location information to other sensors. Then, each sensor node uses the information collected from its neighbour to construct a polygon. Each point in a given Voronoi polygon is closer to the sensor node in this polygon than to any other sensor node. To construct a Voronoi polygon, sensor nodes have to first calculate the bisectors of their neighbours and themselves. Each sensor, represented by a dot, is enclosed by a Voronoi polygon. These Voronoi polygons altogether cover the entire monitoring field. The points inside one Voronoi polygon are much closer to the sensor inside that polygon than other sensors positioned elsewhere. Every sensor is expected to detect any expected event in its own polygon only, not other sensors. Therefore, to detect coverage holes, every sensor is required to check its own polygon only. If the sensor sensing area cannot cover its polygon, that means there are some coverage holes.

4.3. Calculation of New Candidate Location. To calculate the new candidate position in a local polygon, each sensor node will now check its local polygon for any possible coverage holes. If there exists any coverage hole, the sensor node finds a new candidate location for itself by using EBC method, so by relocating there it will eliminate or reduce the size of the coverage hole. The local coverage of local Voronoi polygon is calculated by the intersection of the polygon and the sensing circle. To determine the existence of local coverage hole (LCH) in each polygon, three possible cases of LCH occurrence are considered:

$$\text{LCH} = \begin{cases} P_A - \text{SR}_A & \text{Case 1} \\ 0 & \text{Case 2} \\ P_A - (P \cap \text{SR})_A & \text{Case 3.} \end{cases} \tag{1}$$

```
(1) Find the intersection point between polygon and sensing range
(2) Construct a new polygon based on the old polygon vertices that
      are located inside the SR and the intersection points
(3) Find the area of the new constructed polygon
(4) Find the area of circle segments located inside the polygon (if any)
```

Algorithm 1: For calculating polygon and sensing range intersection.

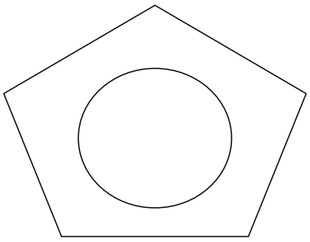

Figure 2: Sensing range fully inside polygon.

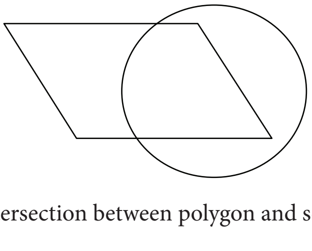

Figure 4: Intersection between polygon and sensing range.

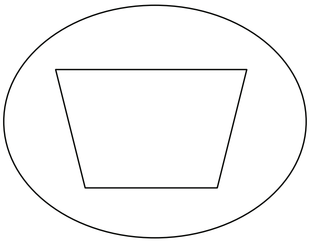

Figure 3: Sensing range fully covered the polygon.

Case 1 ($P_A -$ SR_A). Sensing range is fully included inside polygon as shown in Figure 2: coverage hole = polygon area (P_A) – sensing range area (SR_A).

Figure 2 illustrates the case in which the entire sensing range is covered by the polygon. In this case, it can be seen that some area of the polygon is uncovered which gives a coverage hole in that region. So the area of the polygon and sensing range inside the polygon will be calculated, and then the area of the sensing range is subtracted from the area of polygon.

Case 2 (coverage hole = 0). Polygon is fully covered by the sensing range as illustrated in Figure 3: coverage hole = 0.

Case 3 ($P_A -$ $(P \cap \mathrm{SR})_A$). Polygon is partially covered by the sensing range, meaning that there is an intersection between sensor node sensing range and the polygon as shown in Figure 4: coverage hole = polygon area (P_A) – area of intersection between polygon and sensing range $(P \cap \mathrm{SR})_A$.

Figure 4 is an illustration of the case in which sensing range and the polygons are intersecting each other. So, due to that, intersection of some part of the polygon is covered by the sensing range while another part is not covered resulting in a coverage hole. However, due to the fact that we have to calculate the exact covered area from the polygon and the uncovered region which is the coverage hole too, we start by calculating area of polygon and then subtract the covered area from it; this will give the local coverage hole (LCH) of that polygon as given in

$$\text{LCH} = \text{Area of Polygon} - \text{Covered Area}. \tag{2}$$

The covered polygon area by sensing range has to be calculated by subtracting it from the total polygon area. In the intersection region, a new polygon will be constructed from the big polygon to enable us to calculate the exact covered area. In constructing it, some segments may be created pending on the polygon type and sensing range area that is inside the polygon. Equation (3) represents how the covered area is being calculated:

$$\begin{aligned}\text{Covered Area} = {} & \text{Area of new polygon} \\ & + \sum \text{Area of Segments}.\end{aligned} \tag{3}$$

Algorithm 1 calculates polygon and sensing range intersection in each local polygon so as to determine covered and uncovered areas, so a new candidate location to relocate the sensor node can be calculated.

4.3.1. Calculating New Candidate Location with EBC Algorithm. The EBC algorithm fixes coverage hole by relocating sensor node to the centroid of the polygon as shown in Figure 5. In finding the candidate location, each sensor calculates the length of all edges of the polygon (V_i). Then, the midpoint (MP) of each edge (Eij) in its polygon is calculated and then recorded. Iteratively, assume a polygon such that its vertices are the calculated midpoints. The steps will be stopped when the vertices are very close to each other. Then, the sensor node is relocated to the new location (NL) from the current location (CL) as illustrated in Figure 5, which represents the EBC technique.

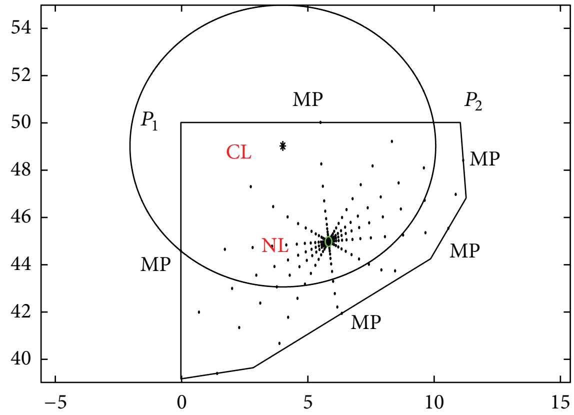

Figure 5: EBC sensor deployment.

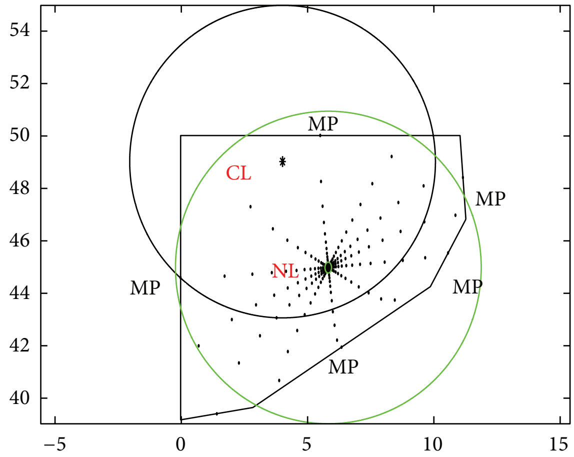

Figure 6: Current and new candidate location.

4.3.2. Polygon Edge Midpoints Calculation. To find the exact midpoint (MP) of polygon edges P_1 and P_2 (assuming that $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$), that is, dividing the edge into two equal parts or halves, we use

$$\mathrm{MP} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right). \qquad (4)$$

4.3.3. Total Coverage of Monitoring Field Calculation. The total area coverage of the monitoring field is after calculating the local coverage of each Voronoi polygon in the sensing field of the partitioned monitoring field. The total coverage of the entire field is presented in

$$\text{Total Coverage} = \mathrm{MFL}_{\text{area}} - \sum_{i=1}^{N} \mathrm{LCH}_i, \qquad (5)$$

where MF is the area of monitoring field, N is the number of sensor nodes, and LCH is local coverage holes of polygon i. Algorithm 2 represents the full EBC algorithm for calculating new candidate location.

4.4. Sensor Node Relocation. When the new candidate location has been calculated, with respect to the new location obtained for sensor node to relocate, the current location will be compared to the new location if there is an increase in area coverage of the sensor node that moved or else maintained its current position. Figure 6 shows the current location of the deployed sensor node and the new candidate location to be moved if the covered area of new candidate location is greater than the current local covered area. The current location is the initial deployment position of sensor node. The previous location of each sensor node is always recorded in order to avoid or control oscillation. The moving oscillation may occur when a new hole is generated due to the sensor node leaving its current location. Sensor nodes are not allowed to go back to their previous location as a control measure. Before a sensor node moves, we first check and compare the newly calculated candidate position with the previous position from the recorded position history. If the new candidate position is the same as the previous candidate position, then the sensor will not move and it will maintain the current position, and direction of movement is opposite to the previous one.

4.5. Termination of Algorithm. In this section, we have the criteria for terminating the program iteration. In this research work, two termination criteria are considered. Firstly, if there is no increment in the local area covered by sensor nodes in its local polygon with 1% or more. The increment is relative to the previous local coverage; always the current calculated area coverage will be compared with the previous one to ascertain the increment. Secondly, if n number of iterations is reached, which is $n = 30$ rounds in our own case, the iteration will be halted or stopped. The complete EBC movement assisted sensor algorithm is presented in Algorithm 3.

5. Simulation and Results

In this section, we present the simulation result. The implementation of the proposed method is carried out in Matlab (version R2011b), on a flat surface without any obstacle of dimension 50 m × 50 m. The total number of sensors (n) deployed randomly varies between $n = 20, 30, 40,$ and 50 with sensing range 6 m and communication range 20 m. In these simulations, the termination condition of the algorithms is when there is no area coverage increment in a local Voronoi polygon by more than 1% for the next move; otherwise, it will maintain current location. There are 30 rounds for each run; the experiments were run 100 times based on different initial random distribution of sensor nodes and the average values of results are calculated. The considered energy consumed to move a sensor node 1 m distance without any stop is 8.268 J [39–41].

The proposed EBC algorithm is compared with Maxmin-Vertex, Maxmin-Edge, Minmax-Edge, Minimax, VOR, and centroid algorithms in which common features among them are used in calculating sensor node new candidate location. Three different criteria are used to evaluate the performance of the EBC method against other existing algorithms under the same distributed networks system such as sensors deployment quality, energy consumption, and network convergence rate.

```
Step 1. For each $V_i$
Step 2.  Repeat
Step 3.      For each Edge
Step 4.          Find the Mid-Point (MP) of Edge
Step 5.          Record the MP
             end for Edge
Step 6.                          Consider a new polygon where the recorded MPs as Vertices
Step 7. Stop when vertices are very close to each other
Step 8. Considered the Candidate position as the statistical average of the vertices
Loop to next polygon
```

Algorithm 2: Edge Based Centroid (EBC) of *i*th Voronoi polygon (V_i).

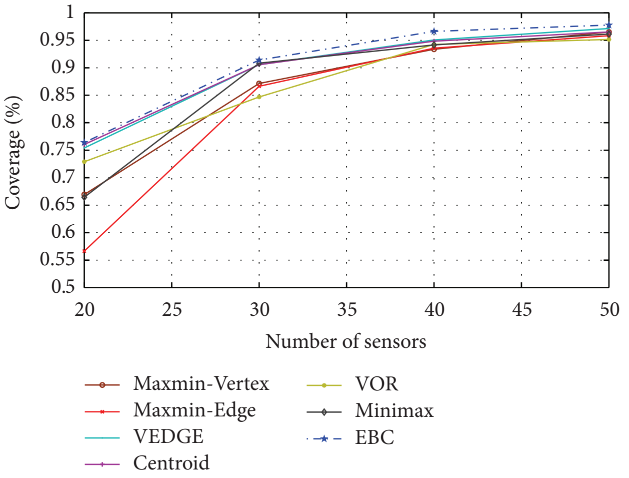

Figure 7: Area coverage percentage.

5.1. Sensors Deployment Quality. The deployment quality of sensor network is the area coverage by deployed sensor nodes. Sensor nodes relocate from one location to another in order to enhance their area coverage. The coverage is calculated in two steps: each sensor first calculates its local Voronoi polygon coverage, then followed by the summation of all the calculated local coverage values in the monitoring field. Figure 7 shows the coverage percentage of the proposed method and other existing methods. The final results obtained are the average values of 100 runs of each algorithm with different initial random deployment. This reveals that coverage of the sensing field increases as the number of sensor nodes increases. Comparing EBC algorithm and other existing methods, VEDGE, Maxmin-Vertex, Maxmin-Edge, Minmax-Edge, Minimax, and VOR, the results reveal that EBC has higher area coverage rising from 76% with 20 sensors to 98% with 50 sensors as shown in Figure 7. The results reveal that all the methods perform very well with the total area coverage of above 90%. Finally, the results reveal that the proposed EBC algorithm outperforms existing methods when compared in terms of area coverage. This is due to the fact that the algorithms always ensure that the new candidate location to be moved is greater than the current location.

5.2. Energy Consumption. The energy consumption is defined based on a sensor node local movement to a certain distance in order to minimize or eliminate a coverage hole. The movement energy consumed in the proposed EBC algorithm and existing methods is measured or calculated to determine how energy has been consumed by each of the algorithms. The consumed energy in relocating a sensor node to *n* meters is in two parts as in [29]: starting/braking energy and moving distance energy. For braking or stopping, the energy needed to overcome the static friction in order to move it again is also the same as the energy value to move 1 m distance as in [42]. The moving distance can be calculated as cumulative energy for the algorithm to complete the 30 rounds or the energy consumed per round. We considered the cumulative energy consumed for 30 rounds. The final cumulative consumed energy results obtained are the average values of 100 runs of each algorithm with different initial random deployment as shown in Figure 8. The results show that moving more sensor nodes or covering far distance by few sensor nodes can lead to higher energy consumption. Comparing the proposed EBC algorithm with current works, the results reveal that EBC has lower energy consumption in three scenarios ranging from 708 joules to 888 joules, except in one scenario (last scenario with 50 sensor nodes) of VOR method with 627 joules and EBC with 708 joules. The low energy consumption recorded by VOR algorithm in that scenario is due to lower area coverage of 95% achieved compared to EBC with 98%, which reveals that there is more sensor movement in EBC to achieve that coverage than VOR. The presented result in Figure 8 finally reveals that EBC algorithm outperformed the existing methods in terms of energy consumption. This is because the distance travelled by most of the sensor nodes to their new candidate location is not much, and movement takes place only if there is an increment in the area coverage.

5.3. Convergence Rate. The convergence rate is presented as the number of rounds in which a method reaches a stage in which there is no increment in area coverage based on a certain threshold value. To determine the convergence rate, we calculate the total area coverage of the covered area at each round and checked whether the area coverage increased with a threshold value of 1% or more and then we move to the next round; otherwise, the algorithm stopped at that round. Then, the number of rounds to reach convergence is recorded. Comparing the proposed EBC algorithm with existing works, the results in Figure 9 reveal that EBC algorithm has the

```
Input:
MFL = integer // Length of Monitoring field
Sensors = integer // Number of Sensor nodes deployed
MR = integer // Maximum number of rounds
Srange = integer // Sensing range radius
Scom = integer // Sensor communication range
PermeterEnergy = 8.268 // Energy consumed per 1 meter
Variables
Positions = [ ] // Array variable of size (Sensors, 2) to record the positions
totalcov = 0 // To record the calculated coverage
totalcovpercent = 0 // To record the calculated coverage percentage
Round = 1 // T count number of rounds
totalEnergyCons = 0 //To Record total energy consumed
VD = [ ] // Array to save voronoi diagram
Start
      newpositions = [ , ] // Array of new positions
      Positions = rand (sensors, 2) // Generating random initial positions
      VD = Voronoi (positions) // Voronoi generation with the generated position
      totalcov = CalculateCoverage (VD, positions) // Function to calculate total coverage
      totalcovpercent = (totalcov/MFL^2) * 100 // calculating total coverage percentage
        For each polygon in VD
        Repeat
              Newpolygon = [ , ]
                 For each Edge in polygon
                   Find midpoint of Edge
              Append midpoint to new polygon
              End For each Edge
           Until vertices are very close to each other
      Record the statistical average of new candidate location into newpositions
        End For each polygon
      Checkforbettercoverage (VD, positions, newpositions) // Function for if NCL > CL
        Newcov = calculatecoverage (VD, newpositions)
        Newcovpercent = (newcov/MFL^2) * 100
        IF coverage enhanced then
        IF Round ≤ MR
        calcEnergy (positions, newposition, permeterEnergy)
              position = newpositions
              totalcov = newtotalcov
              totalcovpercent = newtotalcovpercent
           else
              exit Loop
        End IF Round
        Else
           Exit Loop
        End IF coverage
        Next Round
        calculatecoverage average for all rounds
        calculateEnergy consumption average for all rounds
        calculateconvergence average for all rounds
```

Algorithm 3: Edge Based Centroid (EBC) of *i*th Voronoi polygon (V_i).

lowest convergence rate ranging from two rounds to four rounds. This happened due to the fact that the area coverage increment in each round in EBC algorithm is less than the threshold value (1%).

6. Conclusion and Future Work

This paper proposed Edge Based Centroid (EBC) algorithm that enhanced area coverage of monitoring field with minimal energy consumption in a mobile sensor network iteratively. The proposed EBC algorithm is based on Voronoi diagram that partitions the sensing field into polygons with one sensor node each to monitor any event in its respective subregion. The mobile sensors are moved to new calculated candidate location that is at the center position of each polygon from its current or initial location, which improves area coverage as shown by the results. Using this algorithm for sensor nodes relocation, a set of rules were applied to

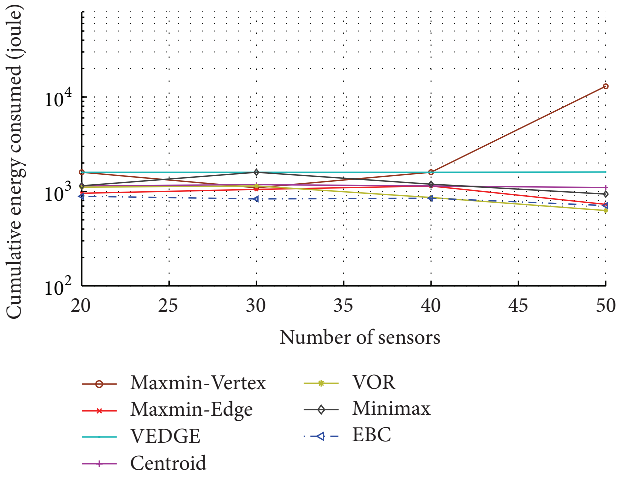

Figure 8: Cumulative energy consumption.

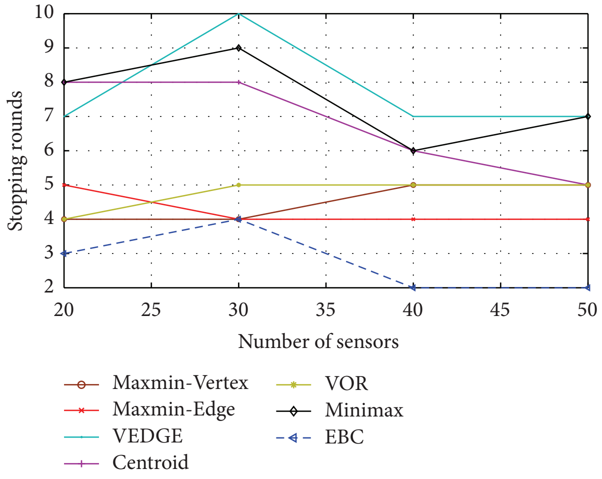

Figure 9: Convergence rate.

ensure that the center of each polygon is calculated before moving the nodes. In terms of movement energy consumption by sensor nodes, the simulation results reveal improved performance of the proposed EBC algorithm in different scenarios.

Considering the scope of this paper which is based on distributed network, real data delivery model is not considered. In future work, the performance of EBC algorithm will be analyzed in a centralized solution that has a sink node. The practical implementation of the proposed EBC algorithm on real hardware devices is needed so as to observe the real behavior of the algorithm. We aim to develop a small test bed for practical implementation of the proposed algorithm on real hardware devices and evaluate its results.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for funding this research. The research is supported by Ministry of Education Malaysia (MOE) and conducted in collaboration with Research Management Center (RMC) at Universiti Teknologi Malaysia (UTM) under VOT no. R.J130000.7828.4F708.

References

[1] M. S. Aliyu, H. Chizari, and A. H. Abdullah, "Integrated sensor and RFID network node placement in disaster monitoring applications," in *Proceedings of the IEEE International Conference on RFID-Technologies and Applications (RFID-TA '13)*, pp. 1–5, Johor Bahru, Malaysia, September 2013.

[2] Y. Zhang, X. Zhang, Z. Wang, and H. Liu, "Virtual edge based coverage hole detection algorithm in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '13)*, pp. 1488–1492, Shanghai, China, April 2013.

[3] M. Liu, H. Gong, Y. Wen, G. Chen, and J. Cao, "The last minute: efficient data evacuation strategy for sensor networks in post-disaster applications," in *Proceedings of the IEEE INFOCOM*, pp. 291–295, Shanghai, China, April 2011.

[4] T. Ma, Y. Liu, J. Fu, and Y. Jing, "Forest fire monitoring based on mixed wireless sensor networks," *International Journal of Smart Home*, vol. 9, no. 3, pp. 169–184, 2015.

[5] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 174–185, Seattle, Wash, USA, 1999.

[6] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[7] L. Mo, W. Peng-Jun, and O. Frieder, "Coverage in wireless ad hoc sensor networks," *IEEE Transactions on Computers*, vol. 52, pp. 753–763, 2003.

[8] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Ga, USA, 2002.

[9] J. Liang, M. Liu, and X. Kui, "A survey of coverage problems in wireless sensor networks," *Sensors & Transducers*, vol. 163, no. 1, pp. 240–246, 2014.

[10] M. Abbasi, M. S. Bin Abd Latiff, and H. Chizari, "Bioinspired evolutionary algorithm based for improving network coverage in wireless sensor networks," *The Scientific World Journal*, vol. 2014, Article ID 839486, 8 pages, 2014.

[11] A. Howard, M. J. Mataríc, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.

[12] Y. Kim, M. Yeo, D. Kim, and K. Chung, "A node deployment strategy considering environmental factors and the number of nodes in surveillance and reconnaissance sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 385282, 11 pages, 2012.

[13] G. Tuna, V. C. Gungor, and K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots

for human existence detection in case of disasters," *Ad Hoc Networks*, vol. 13, pp. 54–68, 2014.

[14] D. Wang, J. Liu, and Q. Zhang, "Mobility-assisted sensor networking for field coverage," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1190–1194, Washington, DC, USA, November 2007.

[15] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

[16] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing ranges," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–6, Miami, Fla, USA, December 2010.

[17] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, pp. 345–405, 1991.

[18] S. Fortune, "Voronoi diagrams and Delaunay triangulations," *Computing in Euclidean geometry*, vol. 1, pp. 193–233, 1992.

[19] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, 1998.

[20] C. Xu, X. Bai, L. Ding, J. Teng, S. Chellappan, and D. Xuan, "Directed coverage in wireless sensor networks: concept and quality," *Ad-Hoc & Sensor Wireless Networks*, vol. 11, no. 3-4, pp. 173–197, 2011.

[21] S. Sendra, J. Lloret, C. Turró, and J. M. Aguiar, "IEEE 802.11a/b/g/n short-scale indoor wireless sensor placement," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1–3, pp. 68–82, 2014.

[22] J. Lloret, J. J. López, C. Turró, and S. Flores, "A fast design model for indoor radio coverage in the 2.4 GHz wireless LAN," in *Proceedings of the 1st International Symposium on Wireless Communication Systems (ISWCS '04)*, pp. 408–412, Port Louis, Mauritius, September 2004.

[23] A. Hawbani and X. Wang, "Zigzag coverage scheme algorithm & analysis for wireless sensor networks," *Network Protocols and Algorithms*, vol. 5, no. 4, pp. 19–38, 2013.

[24] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 163–174, 2014.

[25] J. Li, B. Zhang, L. Cui, and S. Chai, "An extended virtual force-based approach to distributed self-deployment in mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 417307, 14 pages, 2012.

[26] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed coverage and exploration in unknown non-convex environments," in *Distributed Autonomous Robotic Systems*, A. Martinoli, F. Mondada, N. Correll et al., Eds., vol. 83, pp. 61–75, Springer, Berlin, Germany, 2013.

[27] H. Mahboubi, J. Habibi, A. G. Aghdam, and K. Sayrafian-Pour, "Distributed deployment strategies for improved coverage in a network of mobile sensors with prioritized sensing field," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 451–461, 2013.

[28] H.-J. Lee, Y.-H. Kim, Y.-H. Han, and C. Y. Park, "Centroid-based movement assisted sensor deployment schemes in wireless sensor networks," in *Proceedings of the IEEE 70th Vehicular Technology Conference Fall (VTC-Fall '09)*, pp. 1–5, Anchorage, Alaska, USA, September 2009.

[29] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[30] H. Chizari, M. Hosseini, T. Poston, S. A. Razak, and A. H. Abdullah, "Delaunay triangulation as a new coverage measurement method in wireless sensor network," *Sensors*, vol. 11, no. 3, pp. 3163–3176, 2011.

[31] F. Zhao and L. J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, Boston, Mass, USA, 2004.

[32] Z. Yu, J. Teng, X. Bai, D. Xuan, and W. Jia, "Connected coverage in wireless networks with directional antennas," *ACM Transactions on Sensor Networks*, vol. 10, no. 3, article 51, 2014.

[33] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 139–150, Rome, Italy, July 2001.

[34] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[35] A. Boukerche and F. Xin, "A voronoi approach for coverage protocols in wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 5190–5194, Washington, DC, USA, November 2007.

[36] H. Chizari, T. Poston, S. Abd Razak, A. H. Abdullah, and S. Salleh, "Local coverage measurement algorithm in GPS-free wireless sensor networks," *Ad Hoc Networks*, vol. 23, pp. 1–17, 2014.

[37] H. Chizari, T. Poston, S. Abd Razak, A. H. Abdullah, and S. Salleh, "Local coverage measurement algorithm in GPS-free wireless sensor networks," *Ad Hoc Networks*, vol. 23, pp. 1–17, 2014.

[38] A. Kwok and S. Martínez, "A distributed deterministic annealing algorithm for limited-range sensor coverage," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 792–804, 2011.

[39] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 19–24, Taipei, Taiwan, September 2003.

[40] S. Yoon, O. Soysal, M. Demirbas, and C. Qiao, "Coordinated locomotion and monitoring using autonomous mobile sensor nodes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1742–1756, 2011.

[41] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 163–175, 2014.

[42] G. Wang, G. Cao, P. Berman, and T. F. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 515–528, 2007.