
Volume 30, issues 14–15, 15 October 2007
ISSN 0140-3664
ELSEVIER
computer
communications
Available online at
ScienceDirect
www.sciencedirect.com
Special Issue:
Network Coverage and Routing Schemes for Wireless Sensor Networks
Guest Editors: Hsiao-Hwa Chen
Yang Yang

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Computer Communications 30 (2007) 2744–2752

computer communications

www.elsevier.com/locate/comcom

A Delaunay Triangulation based method for wireless sensor network deployment

Chun-Hsien Wu, Kuo-Chuan Lee, Yeh-Ching Chung *

Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan, ROC

Available online 15 June 2007

Abstract

To obtain a satisfied performance of wireless sensor network, an adaptable sensor deployment method for various applications is essential. In this paper, we propose a centralized and deterministic sensor deployment method, DT-Score (Delaunay Triangulation-Score), aims to maximize the coverage of a given sensing area with obstacles. The DT-Score consists of two phases. In the first phase, we use a contour-based deployment to eliminate the coverage holes near the boundary of sensing area and obstacles. In the second phase, a deployment method based on the Delaunay Triangulation is applied for the uncovered regions. Before deploying a sensor, each candidate position generated from the current sensor configuration is scored by a probabilistic sensor detection model. A new sensor is placed to the position with the most coverage gains. According to the simulation results, DT-Score can reach higher coverage than grid-based and random deployment methods with the increasing of deployable sensors.

Keywords: Wireless sensor network; Sensor deployment; Sensor coverage; Obstacles; Delaunay Triangulation

1. Introduction

Wireless sensor network (WSN) is one of the key elements to the success of pervasive/ubiquitous computing. With the advance of wireless communication, system-on-chip (SoC), and micro-electro-mechanical systems (MEMS), the hardware infrastructure of wireless sensor network is getting more mature. Many feasible applications are proposed such as industrial sensor networks [12], volcano monitoring networks [20], habitat monitoring [21], health monitoring [21], and home automation [21], etc.

To obtain a satisfied performance of wireless sensor network, an adaptable sensor deployment method for various applications is essential. The degree of sensor coverage is a major performance metric of sensor deployment method. Sensor coverage can be categorized into three types: area coverage, point coverage, and barrier coverage [4]. For area coverage, sensors have to cover all of the sensing area. If the number of sensors is not sufficient to ensure full coverage, coverage holes will appear [1]. For point coverage, a set of target points must be covered by sensors. For barrier coverage, the goal is to minimize the probability of undetected objects pass through the barrier formed by wireless sensor networks.

In this paper, a centralized and deterministic sensor deployment method, DT-Score (Delaunay Triangulation-Score), is proposed. Given a fixed number of deployable sensors, DT-Score aims to maximize the area coverage of a sensing area with obstacles. The DT-Score consists of two phases. In the first phase, we use a contour-based deployment to eliminate the coverage holes near the boundary of sensing area and obstacles. In the second phase, a deployment method based on the Delaunay Triangulation is applied for the uncovered regions. Before deploying a sensor, each candidate position generated from the current sensor configuration is scored by a probabilistic sensor detection model. Then a new sensor is placed to the

* Corresponding author. Tel.: +886 3 574 2971; fax: +886 3 572 3694.
E-mail addresses: chwu@cs.nthu.edu.tw (C.-H. Wu), kclee@cs.nthu.edu.tw (K.-C. Lee), ychung@cs.nthu.edu.tw (Y.-C. Chung).

0140-3664/$ - see front matter
doi:10.1016/j.comcom.2007.05.017

position with the most coverage gains. To evaluate the performance of DT-Score, we compare it with a grid-based deployment method, MAX_MIN_COV [8], and a random deployment method. In MAX_MIN_COV, sensors must be placed on the predefined grid points distributed to the whole sensing area. A simulation is conducted for four different scenarios. The results show that the area coverage of DT-Score is better than that of MAX_MIN_COV in most cases. The coverage of MAX_MIN_COV is bounded by the density of grid points. In contrast, the DT-Score can achieve higher coverage as the number of deployable sensor increasing. The DT-Score also outperforms the random deployment method in all scenarios.

The rest of the paper is organized as follows. In Section 2, we briefly describe previous works related to the area coverage of sensor deployment. In Section 3, some backgrounds related to DT-Score are given. In Section 4, we present the details of DT-Score. Section 5 evaluates the performance of DT-Score under various scenarios. Finally, we conclude the paper in Section 6.

2. Related work

In this paper, we focus on area coverage in sensor deployment. In the following, we will briefly describe some related results based on deterministic and stochastic/dynamic algorithms. Besides, some results that utilize Delaunay Triangulation and Voronoi Diagram are also addressed.

For static environment, deterministic deployment is used since the location of each sensor can be predetermined properly. MAX_AVG_COV and MAX_MIN_COV are two grid-based algorithms proposed in [8], in which sensors must be placed on the predefined grid points distributed to the whole sensing area. These algorithms concentrate on average coverage as well as on maximizing the coverage of the most vulnerable grid points. MAX_AVG_COV tries to place sensors such that the average coverage of grid points will be maximized. In MAX_MIN_COV, the coverage of grid point that is less covered will be maximized. Case studies for sensing area with obstacles and preferential coverage show that the MAX_AVG_COV and MAX_MIN_COV algorithms significantly outperform random and uniform deployment algorithms. In [3], a variety of regular deployment topologies were presented and analyzed, including circular and star deployments as well as deployments in square, triangular, and hexagonal grids. In [15], it considers an unreliable wireless sensor grid-network with nodes placed in a square of unit area. They derived sufficient and necessary conditions for the relations between coverage, connectivity and diameter. In [19], the sensing area was presented as an arbitrary-shaped polygon possibly with obstacles. The sensing area is partitioned into smaller sub-regions based on the shape of the area, and then the sensor is deployed to these regions systematically. This approach assumes that each sensor has predictable communication range and sensing range, and it allows an arbitrary relationship between them.

The stochastic deployment is used when the information of sensing area is not known in advance or is varied with time, that is, the position for sensor deployment cannot be determined. In addition, the positions of deployed sensors need to be adjusted to fix coverage holes. In [17], the Voronoi diagram was used to discover the existence of coverage holes. To construct the Voronoi diagram, it assumes that each sensor knows the location of its neighbors. The sensing area will be partitioned into Voronoi polygons and each polygon contains only one sensor. If the sensing region of a sensor cannot cover the corresponding polygon, the coverage holes will appear. To improve the coverage, movement-assisted sensor deployment protocols are proposed to eliminate coverage holes. In [10], a potential-field-based approach was proposed. Each sensor is regarded as a virtual particle, and virtual forces are generated due to the potential fields between sensors and obstacles or other sensors. This approach does not require environment information of sensing area and communication between sensors. It relies on each sensor that has the ability to detect the range and direction of neighborhood sensors and obstacles. In [23], a virtual force algorithm (VFA) was proposed as a practical approach for sensor deployment. The VFA algorithm used a forced-directed approach to improve the coverage provided by an initial random deployment.

In addition to discover the existence of coverage holes, Voronoi Diagram and Delaunay Triangulation can be used to determine the maximal breach path (MBP) and the maximal support path (MSP) for a given sensor deployment [13,14]. The MBP (or MSP) corresponds to the worst (or best) case coverage that for any point on the path, the distance to the closest sensor is maximized (or minimized) [4]. As a result, the MBP must pass through the edges of Voronoi Diagram and the MSP must pass through the edges of Delaunay Triangulation. The best and worst case coverage can be categorized to the barrier coverage mentioned previously.

If sensors are densely deployed to cover the whole sensing area, then a full coverage can be reached and has the benefit of redundancy. Researches about multiple degrees of sensor coverage to improve network redundancy were proposed in [18,22]. To take the advantage of the benefit of redundancy [5,16], proposed some sensor scheduling mechanisms to extend the network lifetime. In [11], the coverage problem was modeled as a decision problem and two polynomial-time algorithms were proposed to verify whether each location of the sensing area is covered by at least k sensors (called k-coverage). Algorithm k-UC assumes a uniform circular sensing disk while k-NC assumes a non-disk sensing range for each sensor node. These algorithms only require the location information of each deployed sensor to evaluate the desired multiple coverage. In order to determine which sensing regions are less than k-covered, a central controller is needed to collect coverage information.

3. Preliminaries

3.1. Delaunay Triangulation and Voronoi Diagram

Delaunay Triangulation and Voronoi Diagram are important data structures in computational geometry [2,9]. Delaunay Triangulation is the dual structure of the Voronoi diagram in 2-D plane. It satisfies the empty circle property, that is, for each edge in Delaunay Triangulation, we can find a circle passes through the edge's endpoints without enclosing other points. In Fig. 1(a), we can find the largest empty circle from the Delaunay Triangulation of given sensors. The center of the largest empty circle has the weakest detection probability for current available sensors in Fig. 1(b). In our DT-Score algorithm, a sensor will be placed on the center of the largest empty circle to get most coverage gains.

3.2. The sensor detection model

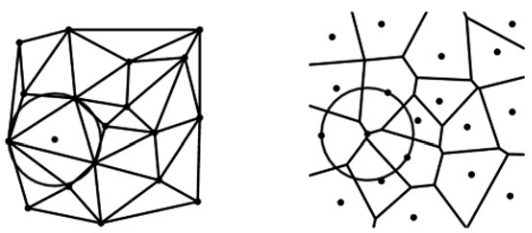
In this paper, we assume that the sensing range of each sensor is a disk with a fixed radius denoted as *SRange*. Assume that a sensor s is deployed at point (x_s, y_s). For any point p at (x_p, y_p), the Euclidean distance between s and p is denoted as $d(s,p)$. A binary sensor model that expresses the coverage rate (or detection probability) of sensor s at point p is given as follows [6]:

$$C_p(s) = \begin{cases} 1, & \text{if } d(s,p) < SRange \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In reality, the sensing range of a sensor is impossible to maintain disk-shaped perfectly. Therefore, a probabilistic sensor detection model of $C_p(s)$ based on Eq. (1) and probabilistic terms in [23] is given as follows:

$$C_p(s) = \begin{cases} 1, & \text{if } d(s,p) \leqslant SRange - PRange \\ e^{-\alpha \times dist^{\beta}}, & \text{if } SRange - PRange < d(s,p) \\ & \leqslant SRange + PRange \\ 0, & \text{if } d(s,p) > SRange + PRange \end{cases} \quad (2)$$

where *PRange* (*PRange* < *SRange*) is defined as the range of uncertainty in sensor detection. If p lies between $SRange - PRange$ and $SRange + PRange$, it may be detected by sensor s, and the detection probability $C_p(s)$ is expressed as $\exp(-\alpha \times dist^{\beta})$, where $dist = (d(s, p) - (SRange - PRange))/2 \times PRange$ is the ratio of $d(s,p)$ within the probabilistic detection range ($2 \times PRange$). The relation between *SRange* and *PRange* is illustrated in Fig. 2. Fig. 3 is the probabilistic sensor detection model for different sensor parameters α and β modified from [23]. By adjusting α and β, this model can be used to express different types of sensor nodes.



(a) Delaunay Triangulation. (b) Voronoi Diagram.

Fig. 1. Delaunay Triangulation and Voronoi Diagram.

4. Sensor deployment based on Delaunay Triangulation

In this paper, we focus on deterministic deployment in a wireless sensor network. Assume that we know the position of each deployed sensor. In order to improve the area coverage and reduce the number of sensors used, it is intuitive to place a new sensor to the sparse region of sensing area. The empty circle property of Delaunay Triangulation provides a way for us to find such region. The DT-Score algorithm consists of two phases:

Phase 1 (contour deployment). This phase consists of initialization step and contour points generation step. In the initialization step, a sensing area environment is initialized base on the configuration file. In the contour points generation step, contour points are generated to eliminate the coverage holes near the boundary of sensing area and obstacles. The purpose of the contour deployment is to eliminate the coverage holes generated by the Delaunay Triangulation method around the boundary and obstacles. With more sensors available, these sensors can be deployed to areas other than boundary and obstacles and improve the coverage of the Delaunay Triangulation method.

Phase 2 (refined deployment). This phase consists of candidate positions generation step, scoring step, and sensor addition step. In the candidate positions generation step, the Delaunay Triangulation is used to find the candidate positions for uncovered regions. In the scoring step, each candidate position is scored by a probabilistic sensor detection model. In the sensor addition step, a sensor is deployed to the position with the most coverage gains. In this phase, these three steps are repeated until the predefined number of deployable sensors is reached. In the following, we will describe each phase of the DT-Score algorithm in details.

4.1. Phase 1 – contour deployment

4.1.1. Initialization step

In this step, a sensing area is generated from a given configuration file. An example of a given configuration file is

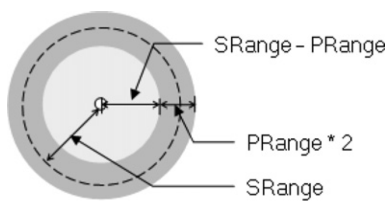


Fig. 2. Sensing range.

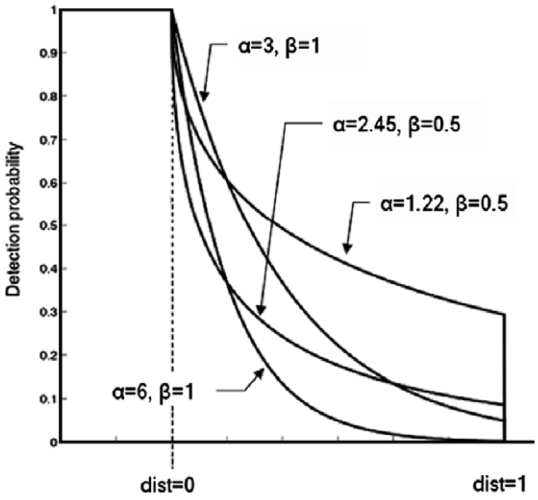


Fig. 3. The probabilistic sensor detection model.

shown in Fig. 4. In Fig. 4, the configuration file includes the size of sensing area, the parameters of sensors for the probabilistic sensor detection model, the position and size for each obstacle (option), and the coordinates of pre-deployed sensors (option). The sensing area is defined as a rectangle and the obstacle is described as a polygon. To give the flexibility of deployment, some pre-deployed sensors can be optionally specified in advance. The position of each sensor will be added to *Sensor* vector.

4.1.2. Contour points generation step

This step is used to eliminate coverage holes near the boundary of sensing area and the obstacles. Initially, contour points are placed alone the boundary of sensing area. In this step, offsets between contour points and boundary are reserved to allow more sensors to be deployed in the next phase. With the proper selection of offsets, we can earn more coverage contributed by contour points. Fig. 5 illustrates the calculation of offset, suppose the radius of sensing region for each sensor is denoted as R, then the offset is R/sqrt(2). To ensure every part in the sensing region can be fully detected by sensor, R is set to *SRange* – *PRange*. The distance between every two adjacent contour points is $2R$/sqrt(2). It ensures that the boundary of sensing area can be fully covered with the least number

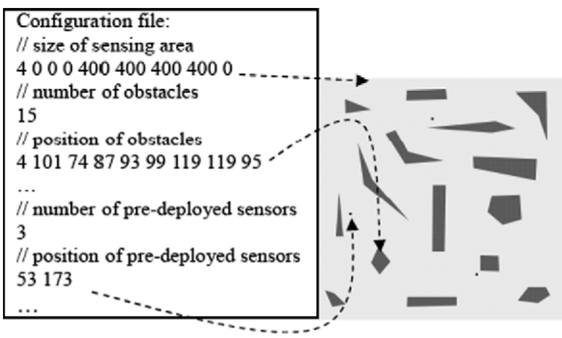


Fig. 4. Initialization step.

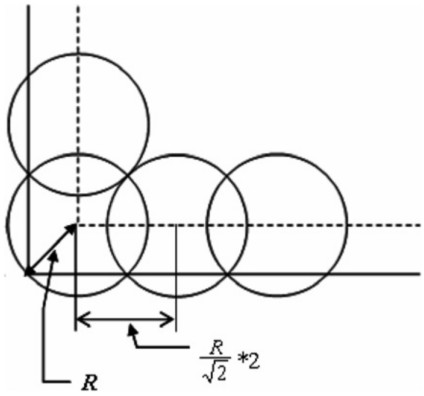


Fig. 5. Offset calculation.

of sensors. The positions of contour points will be added to *Sensor* vector if they are not within any obstacles.

Next, contour points are placed around the obstacles. For each obstacle, we first calculate the line equation of the edges in point–slope form. If the slope of the edge is less than or equal to 1, the contour points are placed with an offset R/sqrt(2) in y-axis away from the obstacle (see Fig. 6(a)). For a vertical edge and the edge with slope greater than 1, the contour points are placed with an offset R/sqrt(2) in x-axis away from the obstacle (see Fig. 6(b)). The distance between any two adjacent contour points is also set to $2R$/sqrt(2). If a contour point is not within any obstacles or outside of the sensing area, it will be added to Sensor vector. Fig. 7 is an example of contour points generation step based on the sensing area shown in Fig. 4.

4.2. Phase 2 – refined deployment

4.2.1. Candidate positions generation step

If the quota of deployable sensors is not used up in the phase 1, a refined deployment method based on the Delaunay Triangulation will be used to improve the coverage. It consists of three steps: candidate generation, scoring, and sensor addition. In the candidate generation step, candidate positions for sensor deployment are generated based on the positions of deployed sensors. In this paper, a centralized, offline approach is used to determine these positions. It is based on the incremental randomized Delaunay Triangulation proposed in [7]. Fig. 8 illustrates the result of applying the Delaunay Triangulation

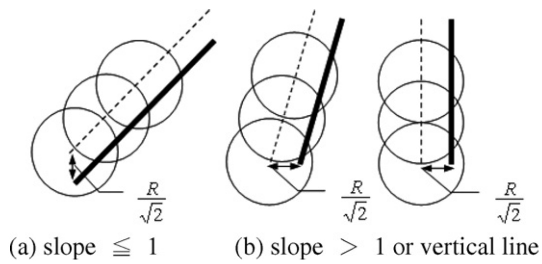


Fig. 6. Offset under different slope.

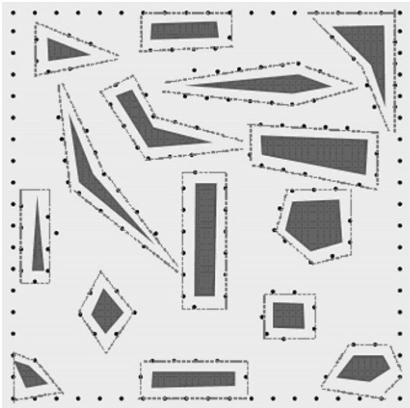


Fig. 7. Contour points generation step.

algorithm to the sensor configuration shown in Fig. 7. According to the empty circle property described in Section 3.1, there are some circumcircles of triangles in Fig. 8 containing no sensors. The centers of these circumcircles are candidate positions of new sensors. Except the positions located on the obstacles, a fixed number of positions will be added to the *Candidate* array according to the radius of circumcircle in decreasing order.

4.2.2. Scoring step

In order to deploy a new sensor with the most coverage gains, a scoring mechanism is used to evaluate each candidate position within *Candidate* array. At first, a grid square is placed and centered on each candidate position. The length of edge is $(SRange + PRange) \times 2$. It ensures that any point within the sensing region is considered. The probabilistic sensor detection model, described in Section 3.2, is used to calculate the coverage gains for a candidate position by summarizing the coverage rates at all grid points. The coverage gain is affected by two factors. The first factor is the ratio of sensing region overlapped with existed sensors. Suppose $r_c = Candidate[i] \cdot radius$ is the radius of circumcircle for a candidate position *i*. If r_c is less than $(SRange - PRange) \times 2$, then the sensing region of sensor on position *i* will overlap with existed sensors. The ratio of non-overlapped sensing region is calculated by the area of gray region in Fig. 9 divided by the area of sensing region with radius $(SRange - PRange)$, where the radius of gray region is $r_c - (SRange - PRange)$. Another factor is the influence of obstacles. If a line that connects a grid point and a candidate position intersects with obstacles, the grid point cannot be detected by a sensor placed on the candidate position and cannot contribute any coverage gains. At last, the score for candidate position *i* is stored in $Candidate[i] \cdot score$. The procedure of scoring step is outlined in Fig. 10.

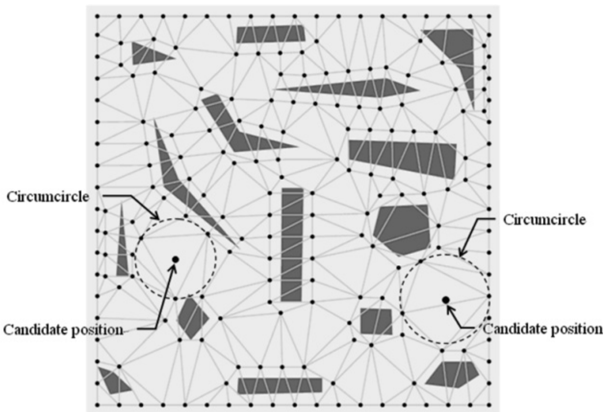


Fig. 8. The Delaunay Triangulation of given sensors.

4.2.3. Sensor addition step

When all candidate positions are scored, the candidate position with the highest score is selected to deploy a new sensor. Thus, the position of new sensor is added to *Sensor* vector. The candidate generation, scoring, and sensor addition steps are repeated until the target number of deployable sensors is reached. Fig. 11 shows the result of the refined deployment phase with 300 sensors. The gray points are sensors newly added in this phase.

The complete procedure of DT-Score algorithm is outlined in Fig. 12. The time complexity of DT-Score algorithm is $O(n^2 \log n)$, where n is the number of sensors and the Delaunay Triangulation algorithm has time complexity of $O(n \log n)$ [7]. For the grid-based deployment methods [8], their time complexity is $O(N^2)$, where N is the total number of grid points in a sensing area. It is clear that the grid-based approach has higher computational overhead than DT-Score when the number of deployable sensors keeps increasing.

5. Performance evaluations

In this section, we evaluate the performance of DT-Score with a grid-based deployment algorithm, MAX_MIN_COV proposed in [8], and a random deployment algorithm modified from DT-Score. We use two grid distances (5 and 10 units) for the MAX_MIN_COV algorithm, denoted as Min-5 and Min-10. The random deployment algorithm has the same initialization step in the first phase of DT-Score, but without the contour points generation step. In the second phase of random deployment, the candidate generation and scoring steps are not

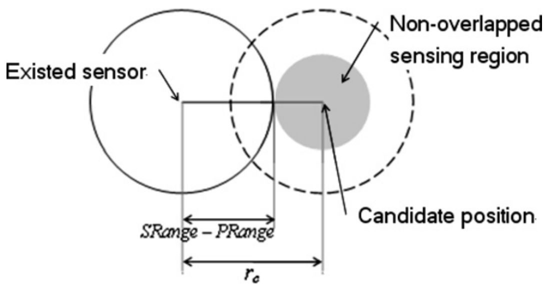


Fig. 9. Overlapping with other sensor.

```
Algorithm Score (Candidate [])
    k = size of Candidate [];
    for ( i=0; i < k; i++ )
    {
        Candidate[i].score = 0;    /* initialize */
        Set a grid rectangle centered on Candidate[i];
        for (all grid points within grid rectangle)
        {
            if (a grid point g is not blocked by any obstacles)
            {
                Candidate [i].score += C_g (i);
                /* coverage rate of sensor i at grid point g */
            }
        }
        if (Candidate [i].radius < (SRange - PRange)*2)
        {
        /* calculate the ratio of non-overlapped sensing region */
            Candidate [i].score *= ((Candidate [i].radius –
                        (SRange - PRange)) / (SRange - PRange))^2;
        }
    }
End_of_Score
```

Fig. 10. Procedure of the Score algorithm.

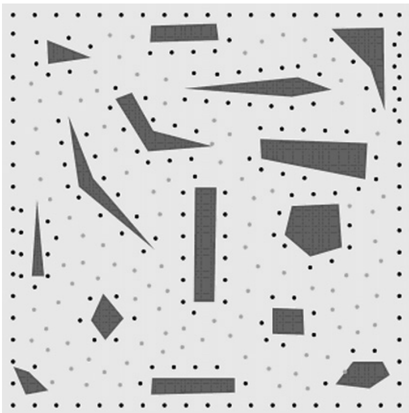


Fig. 11. The result of DT-Score.

```
Algorithm DT-Score ( )
    sensor_num = 0;
    /* initialization step */
    Initialize Obstacle vector with coordinates of obstacles;
    Initialize Sensor vector with coordinates of sensor points;
    /* contour points generation step */
    Add sensors along the boundary of sensing area and the
    edges of obstacles;
    while (sensor_num < limit_num)
    { /* candidates generation step */
      Construct Delaunay Triangulation from Sensor vector;
      for (all circumscribed circles of the triangles)
      {
        Find a center p with the largest radius;
        if (p is not located on any obstacles)
          Add p to Candidate [];
        if (Candidate [] has k elements) break;
      }
    }
    /* candidates scoring and sensor addition step */
    Score (Candidate []);
    Add a candidate with the highest score to Sensor vector;
    sensor_num++;
}
End_of_ DT-Score
```

Fig. 12. Procedure of the DT-Score algorithm.

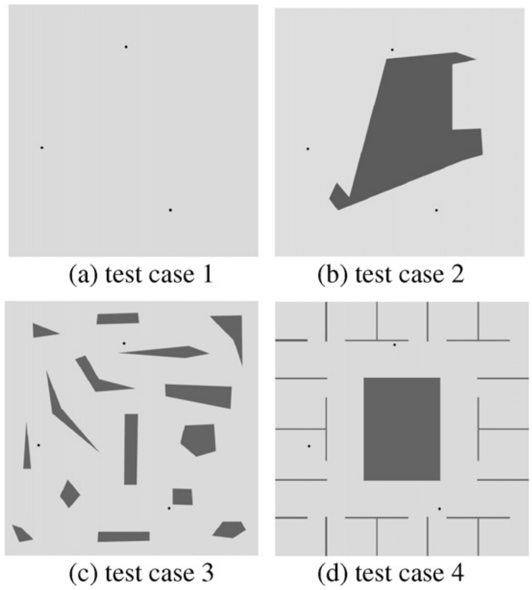


Fig. 13. Maps for test cases.

used. Instead, a candidate position is randomly generated. In the sensor addition step, it only checks if the candidate position is located on any obstacles.

The configurations of the performance evaluations are stated as follows. The sensing area is a 2-D square with 400×400 units. There are four test cases with 0, 1, 15, and 17 obstacles, respectively. The maps of sensing area for each test case are illustrated in Fig. 13. For each test case, three sensor points are deployed in advance. The number of deployable sensors is ranged from 200 to 600. The parameters of sensor are set as $\alpha = 1.22$, $\beta = 0.5$, $SRange = 20$, and $PRange = 5$. To compare the performance between different deployment algorithms, we calculate the coverage of sensing area for each algorithm under different test cases. At first, the sensing area is represented by 400×400 grid points. The coverage is expressed as 1 – *total miss rate of effective grid points*. The "effective" means the grid point not located on any obstacles, and the miss rate of a grid point p can be expressed as 1 – max $(C_p(s))$ for all sensor s, where the $C_p(s)$ is based on the probabilistic sensor detection model described in Section 3.2. For example, if grid point p can be detected by a sensor s without loss ($C_p(s) = 1$), then the miss rate of p is 0.

Fig. 14 is the results of test case 1 for different deployment algorithms. It can be found that DT-Score has higher coverage than the Min-5, Min-10, and random deployment methods when the number of sensors deployed is over 350, 250, and 200, respectively. When the number of deployable sensors is 200, the coverage rate of DT-Score is lower than Min-10. It is because most sensors are deployed in the contour deployment phase of DT-Score. There are insufficient sensors for the phase 2 to improve the coverage. The Min-10 has larger grid distance that reduces the overlap of sensing region for each sensor, and more coverage gains can be earned. For Min-5 and Min-10, we can find that Min-10 has better coverage than Min-5 when the number of sensors deployed is less than 250. It also benefit from the larger grid distance. As a result, we can find that the performance of grid-based deployment is deeply influenced by the density of grid points. When the gird points used over a threshold, the coverage of MAX_MIN_COV reaches to a saturation point. From Fig. 14, we can see that when the number of sensors deployed is over 300 and 250, the coverage of Min-5 and Min-10 will not be increased. For random deployment algorithm, the results are represented with error bars and mean values. We can find that the performance of random deployment is poor than other approaches in most cases. In test case 2, a big obstacle is put on the center of the sensing area. The coverage results are illustrated in Fig. 15. From Fig. 15, we have similar observations as those of Fig. 14.

In test case 3, there are 15 obstacles in the sensing area. From the results illustrated in Fig. 14, we can find that DT-Score is better than other algorithms except for the Min-10 when the number of deployable sensors is 200. It is because that the DT-Score deploys most of the available sensors in the contour deployment phase, and the coverage gains are smaller than the MAX_MIN_COV algorithm with larger grid distance. In Fig. 16, the saturation point of Min-5 and Min-10 is 350 and 250, respectively. In test case 4, there are 17 obstacles in the sensing area. Unlike the various shapes of obstacles in test case 1, the regular-shaped obstacles are used. The results are shown in Fig. 17. We have similar observation as those of Fig. 16. The coverage of grid-based deployment is limited by the density of grid points. DT-Score can achieve higher coverage than MAX_MIN_COV as the number of deployable sensors over a threshold.

6. Conclusions and future work

In this paper, we have proposed a two-phased deterministic strategy, DT-Score, for wireless sensor network deployment. DT-Score is a centralized and deterministic approach that is suitable for planning the position of sensors in the environment with obstacles. In the first phase, the coverage holes near the obstacles and the boundary

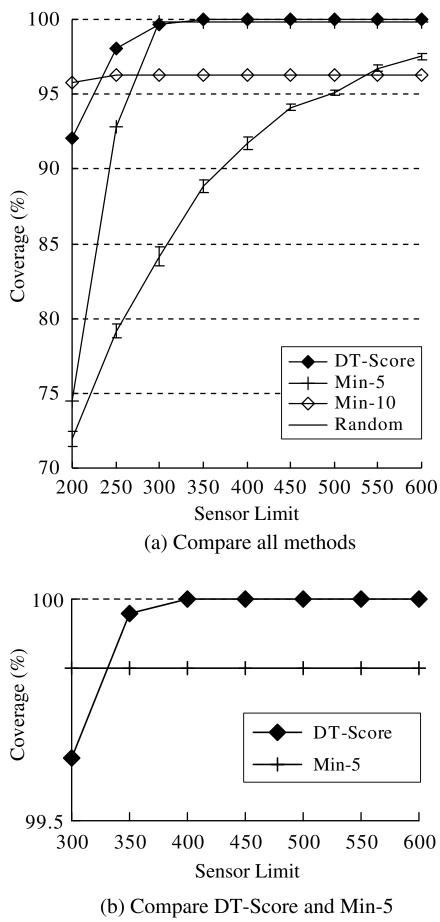


Fig. 14. Results of test case 1.

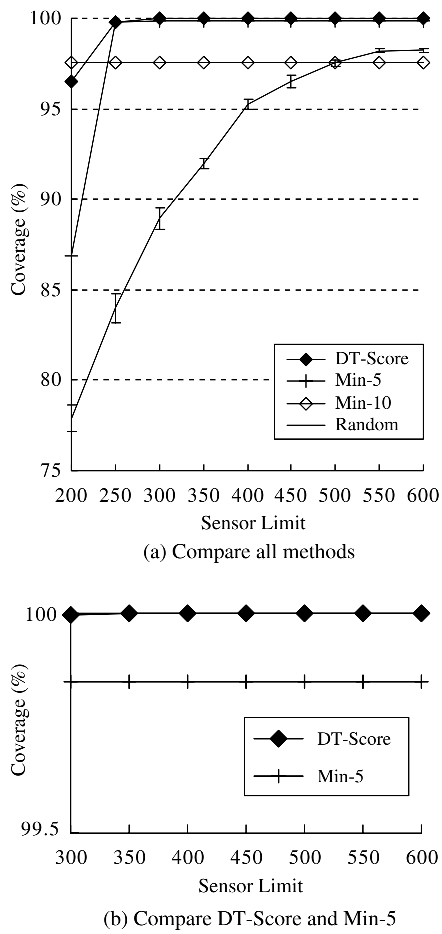
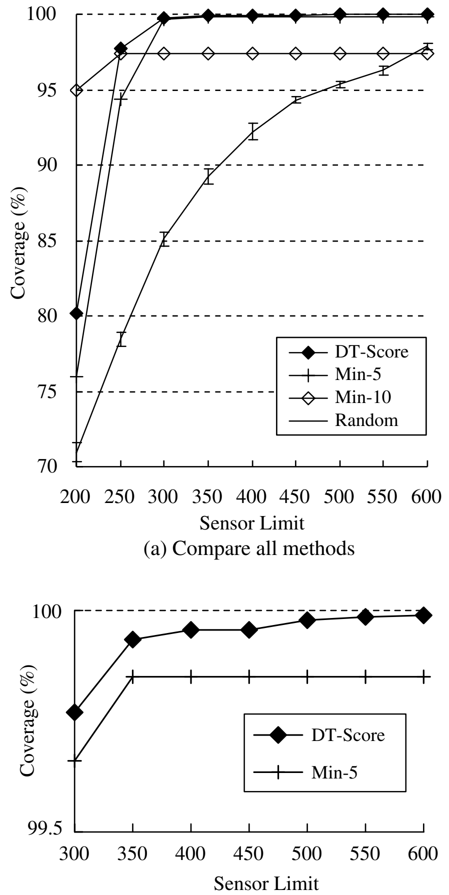


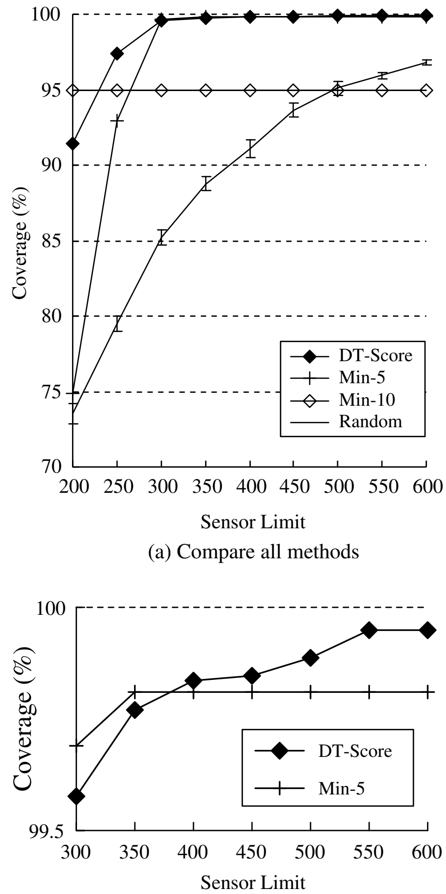
Fig. 15. Results of test case 2.



(a) Compare all methods

(b) Compare DT-Score and Min-5

Fig. 16. Results of test case 3.



(a) Compare all methods

(b) Compare DT-Score and Min-5

Fig. 17. Results of test case 4.

of sensing area are eliminated by using the contour points generation approach. The second phase is a refinement of the result in the first phase. It is based on the Delaunay Triangulation that adds new sensors repeatedly through candidate positions generation and scoring steps. Each candidate position is evaluated by a scoring mechanism based on the probabilistic sensor detection model. This model is more reasonable than the binary detection model, and it is adjustable for different types of sensors. Finally, a candidate position with the most coverage gains will be selected to add new senor.

Compared with grid-based and random deployment approaches, the proposed approach outperforms others as the number of deployable sensors over a threshold. For grid-based approaches, the coverage is limited by the density of grid points. The DT-Score is more scalable than grid-based deployment approaches.

In the future work, the current sensor deployment method will be replaced by online approach. The localization information about obstacles and deployed sensors can be obtained through localization technologies such as GPS. Moreover, a reliable network configuration protocol has to be developed to transfer localization information. In addition to deploy all of the sensors at the same time, new approach should allow to add sensors incrementally during the operation of network based on the current sensor configuration. At least, some characteristics of the wireless sensor network such as different sensor types, more detection and communication models, and irregular sensing region would be addressed.

Acknowledgement

The work of this paper was partially supported by National Science Council and Ministry of Economic Affairs of the Republic of China under contract NSC-94-2213-E-007-036, NSC-94-2213-E-007-037, NSC-94-2752-E-007-004-PAE, and MOEA 94-EC-17-A-04-S1-044.

References

[1] Nadeem Ahmed, Salil Kanhere, Sanjay Jha, The holes problem in wireless sensor networks: a survey, Proceedings of the ACM SIGMOBILE Mobile Computing and Communications Review (MC^{2}R) 9 (2) (2005) 4–18.

[2] F. Aurenhammer, Voronoi Diagrams – A Survey of Fundamental Geometric Data Structure, Proceedings of the ACM Computing Surveys 23 (3) (1991) 345–405.

[3] E.S. Biagioni, G. Sasaki, Wireless sensor placement for reliable and efficient data collection, in: Proceedings of the IEEE 36th Hawaii International Conference on System Sciences (HICSS'03), 2003.

[4] Mihaela Cardei, Jie Wu, in: M. Ilyas, I. Magboub (Eds.), Coverage in wireless sensor networks, Handbook of Sensor Networks, CRC Press, Boca Raton, FL, 2004.

[5] J. Carle, A. Gallais, D. Simplot-Ryl, Preserving area coverage in wireless sensor networks based on surface coverage relay dominating sets, in: Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005), 2005.

[6] K. Chakrabarty, S.S. Iyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks, IEEE Transactions on Computers 51 (2002) 1448–1453.

[7] O. Devillers, Improved incremental randomized Delaunay triangulation, in: Proceedings of the Fourteenth Annual Symposium on Computational Geometry, 1998, pp. 106–115.

[8] S.S. Dhillon, K. Chakrabarty, Sensor placement for effective coverage and surveillance in distributed sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference, 2003, pp. 1609–1614.

[9] D. Du, F. Hwang, S. Fortune, Voronoi diagrams and Delaunay triangulations, Euclidean Geometry and Computers (1992).

[10] A. Howard, M.J. Mataric, G.S. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in: Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems, 2002, pp. 299–308.

[11] Chi-Fu Huang and Yu-Chee Tseng, The coverage problem in a wireless sensor network, in: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, 2003, pp. 115–121.

[12] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, Mark Yarvis, Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea, in: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05), 2005, pp. 64–75.

[13] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B. Srivastava, Worst and best-case coverage in sensor networks, IEEE Transactions on Mobile Computing 4 (1) (2005) 84–92.

[14] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. Srivastava, Coverage Problems in wireless ad-hoc sensor network, in: Proceedings of the IEEE INFOCOM, 2001, pp. 1380–1387.

[15] S. Shakkottai, R. Srikant, N. Shroff, Unreliable sensor grids: coverage, connectivity and diameter, in: Proceedings of the IEEE INFOCOM, 2003, pp. 1073–1083.

[16] D. Tian, N.D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, in: Proceedings of the ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA), 2002.

[17] Guiling Wang, Guohong Cao, Tom La Porta, Movement-assisted sensor deployment, in: Proceedings of the IEEE INFOCOM, 2004, pp. 2469–2479.

[18] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, Christopher Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proceedings of the ACM SenSys'03, 2003, pp. 28–39.

[19] Y.C. Wang, C.C. Hu, and Y.C. Tseng, Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks, in: Proceedings of the Wireless Internet Conf. (WICON), 2005.

[20] Geoffrey Werner-Allen, Konrad Lorincz, Mario Ruiz, Omar Marcillo, Jeff Johnson, Jonathan Lees, Matt Welsh, Deploying a wireless sensor network on an active volcano, IEEE Internet Computing 10 (2) (2006) 18–25.

[21] Ning Xu, A Survey of Sensor Network Applications, University of Southern California. Available from <http://enl.usc.edu/~ningxu/papers/survey.pdf>, 2003.

[22] T. Yan, T. He, J. Stankovic, Differentiated surveillance for sensor networks, in: Proceedings of the ACM SenSys '03, 2003.

[23] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in: Proceedings of the IEEE INFOCOM, 2003, pp. 1293–1303.

Chun-Hsien Wu received the BS degree in applied mathematics from National Chung Hsing University in 2000, and the MS degree in computer science from Feng Chia University in 2002. He is currently a PhD student in the Department of Computer Science at National Tsing Hua University. His research interests are in the areas of wireless sensor network, pervasive computing, embedded system, parallel and distributed computing. He is a student member of the IEEE.

Kuo-Chuan Lee received the BS degree in computer and information science from National Chiao Tung University in 2003, and the MS degree in computer science from National Tsing Hua University in 2005. His research interests are in the areas of wireless sensor network, pervasive computing, and embedded system.

Yeh-Ching Chung received the BS degree in information engineering from Chung Yuan Christian University in 1983, and the MS and PhD degrees in computer and information science from Syracuse University in 1988 and 1992, respectively. He joined the Department of Information Engineering at Feng Chia University as an associate professor in 1992 and became a full professor in 1999. From 1998 to 2001, he was the chairman of the department. In 2002, he joined the Department of Computer Science at National Tsing Hua University as a full professor. His research interests include parallel and distributed processing, pervasive computing, embedded software, and system software for SOC design. He is a member of the IEEE Computer Society and ACM.