A Delaunay Triangulation Based Method for Wireless Sensor Network Deployment

Chun-Hsien Wu, Kuo-Chuan Lee, and Yeh-Ching Chung
Department of Computer Science
National Tsing Hua University, Hsinchu, Taiwan 30013, R.O.C.
{chwu, kclee, ychung}@cs.nthu.edu.tw

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 sensor deployment method, DT-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.

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 [10], volcano monitoring networks [16], habitat monitoring [17], 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 [3]. 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 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 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 [6], 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

Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS'06)
0-7695-2612-8/06 $20.00 © 2006 IEEE

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 [6], 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 [13], 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 [15], 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.

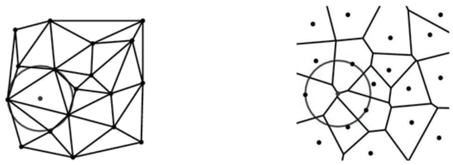
The stochastic deployment is used when the information of sensing area is not known in advance or is varied with time, that is, the position for sensor deployment cannot be determined. In addition, the positions of deployed sensors need to be adjusted to fix coverage holes. In [14], the Voronoi diagram 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 [9], 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 sensor 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 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 [11], [11]. 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) [3]. 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.

3. Preliminaries

3.1. Delaunay Triangulation and Voronoi Diagram

Delaunay Triangulation and Voronoi Diagram are important data structures in computational geometry [2], [8]. 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.



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

Fig. 1. Delaunay Triangulation and Voronoi Diagram.

3.2. The sensor detection model

In this paper, we assume that the sensing range of each sensor is a disk with fixed radius and is 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 of sensor s at point p is given as follows [4]:

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

In reality, the sensing range of a sensor is impossible to maintain a disk shape perfectly. Therefore, a probabilistic sensor detection model of $C_p(s)$ based on Equation (1) and probabilistic terms in [18] is given as follows:

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

where $dist = (d(s, p) - (SRange - PRange))/2*PRange$ is the ratio of $d(s, p)$ within probabilistic detection range $2*PRange$ ($PRange < SRange$). Fig. 2 illustrates the *SRange* and *PRange* used in Equation (2). We can find that if $d(s, p)$ is less than or equal to ($SRange - PRange$), point p can be detected by sensor s without loss. If $d(s, p)$ is larger than ($SRange - PRange$) and less than or equal to ($SRange + PRange$), point p can be detected by sensor s with a probability defined in Equation (2). Fig. 3 is the probabilistic sensor detection model for different sensor parameters α and β modified from [18]. By adjusting these parameters, this model can be used to express different types of sensor such as infrared or ultrasound sensors.

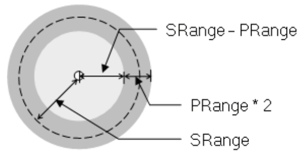


Fig. 2. Sensing range.

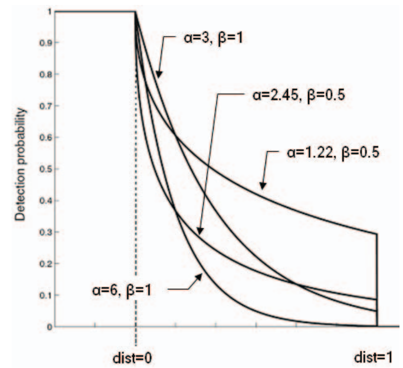


Fig. 3. The probabilistic sensor detection model.

4. Sensor Deployment Based On Delaunay Triangulation

In this paper, we focus on deterministic deployment in 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. The first phase is contour deployment. It consists of initialization step and contour points generation step. In the initialization step, a sensing area environment is initialized base on the configuration file. Next, the contour points are generated to eliminate the coverage holes near the boundary of sensing area and obstacles. The second phase is refined deployment. It 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. The second phase is 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 one – contour deployment

4.1.1. Initialization step. In this step, a sensing area is generated from a given configuration file. This file includes the size of sensing area, the parameters of sensors for the probabilistic sensor detection model, the position and size for each obstacle, and coordinates of pre-deployed sensors. The sensing area is defined as a rectangle and the obstacle is modeled as a polygon.

0-7695-2612-8/06 $20.00 © 2006 IEEE

Sensor vector is used to store the position of deployed sensors. Fig. 4 is an example of initialization step.

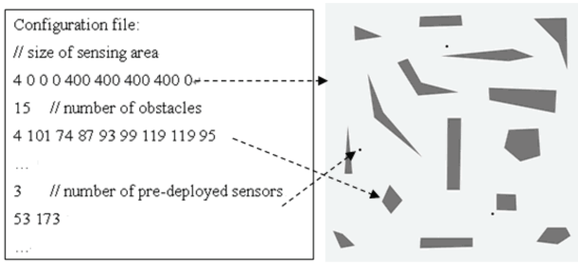


Fig. 4. Initialization step.

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 order to get more coverage gains, an offset is existed between contour point and boundary. Fig. 5 illustrates the calculation of offset, suppose the radius of sensing region for each sensor is denoted to *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 of sensors. The positions of contour points will be added to *Sensor* vector if they are not within any obstacles.

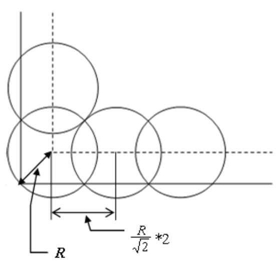
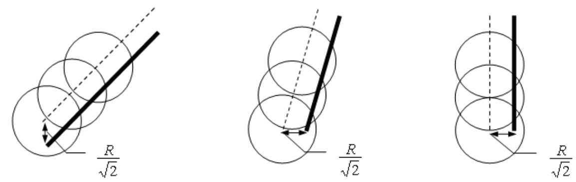


Fig. 5. Offset calculation.

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)). If the slope of the edge is greater than 1 and less than 10, the contour points are placed with an offset *R/sqrt(2)* in x-axis away from the obstacle (See Fig. 6(b)). For the case where vertical edge or the slope of the edge greater than or equal to 10, the contour points are placed with an offset *R/sqrt(2)* in x-axis away from the obstacle (See Fig. 6(c)). 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.



(a) slope <= 1 (b) 1< slope < 10 (c) vertical line or slope >= 10

Fig. 6. Offset under different slope.

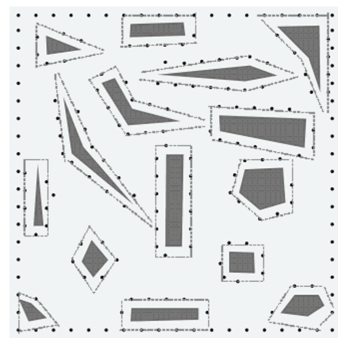


Fig. 7. Contour points generation step.

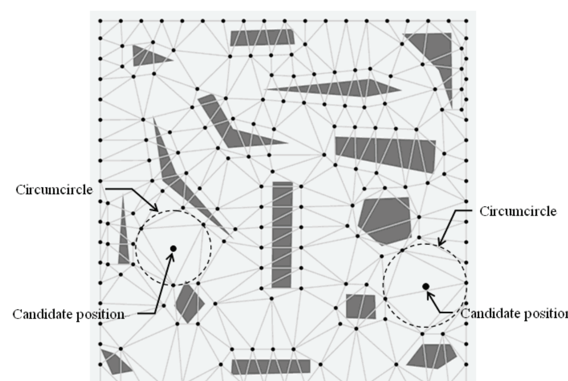


Fig. 8. The Delaunay Triangulation of given sensors.

4.2. Phase two – refined deployment

4.2.1. Candidate positions generation step. In this step, candidate positions for sensor deployment are generated. We use the Delaunay Triangulation algorithm proposed in [5] to determine these positions. Fig. 8 illustrates the result of applying the Delaunay Triangulation algorithm to the sensor configuration shown in the Fig. 7. According to the empty circle property described in Section 3.1, there are some

IEEE COMPUTER SOCIETY

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.

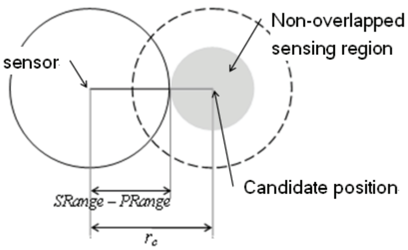


Fig. 9. Overlapping with other sensor.

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*)*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].radius* is the radius of circumcircle for a candidate position *i*. If r_c is less than (*SRange* – *PRange*)*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].score*. The procedure of scoring step is outlined in Fig. 10.

```
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.

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.

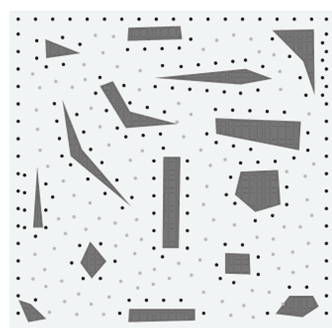


Fig. 11. The result of DT-Score.

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.

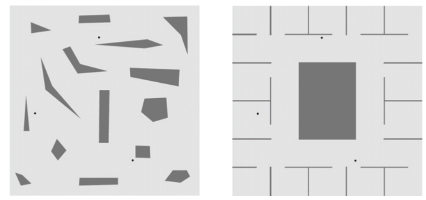
```
Algorithm DT-Score ( )
    sensor_num = 0;
    /* initialization step */
    Initialize Obstacle vector with coordinates of obstacles;
    Initialize Sensor vector with coordinates of sensor points;
    /* contour points generation step */
    Add sensors along the boundary of sensing area and the 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.

5. Performance Evaluations

In this section, we evaluate the performance of DT-Score with a grid-based deployment algorithm, MAX_MIN_COV proposed in [6], 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, the candidate generation and scoring steps are not 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 two test cases with 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 α=1.22, β=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.



(a) test case 1 (b) test case 2

Fig. 13. Maps for test cases

In test case 1, 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. Besides, the characteristic of grid-based deployment is observed as well, that is, as the gird points used up, the coverage of MAX_MIN_COV reaches to a saturation point. In Fig. 14, the saturation point of Min-5 and Min-10 is 350 and 250 respectively. In contrast, Min-5 has much poor coverage than Min-10 if the number of deployable sensors is small. It is because that Min-10 has larger grid distance that reduces the overlap of sensing region, and more coverage gains can be earned. As a result, we can find that the performance of grid-based deployment is deeply influenced by the density of grid points. 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, 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. 15. We have similar observation as those of Fig. 14. 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.

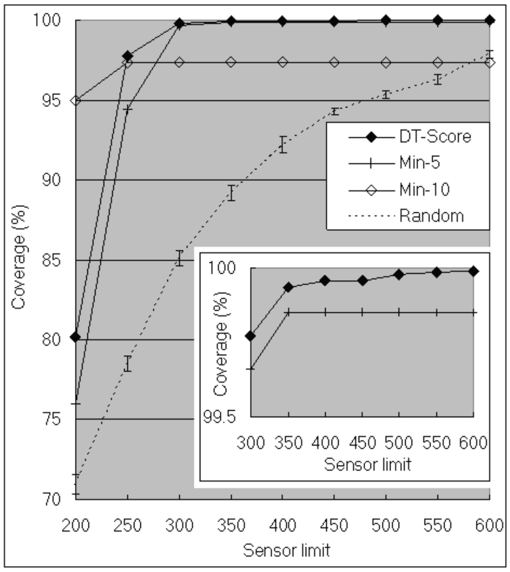


Fig. 14. Results of test case 1.

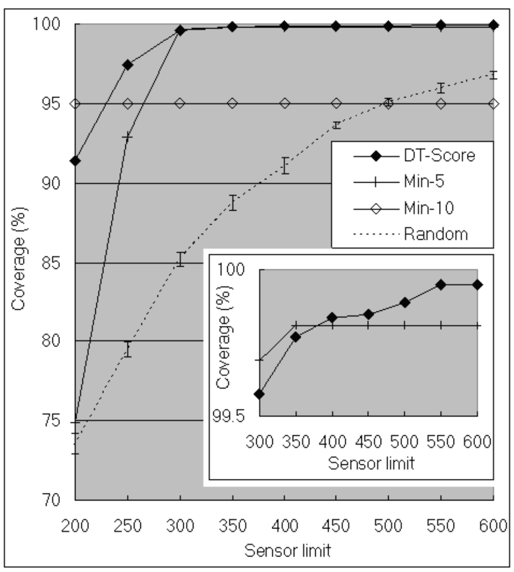


Fig. 15. Results of test case 2.

6. Conclusions

In this paper, we have proposed a two-phased deterministic strategy, DT-Score, for wireless sensor network deployment. DT-Score is a centralized 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 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 real usage, our approach needs the locations of obstacles and pre-deployed sensors. They can be obtained through localization technologies. In addition to deploy all of the sensors at the same time, our approach is allowed to add sensors incrementally during the operation of network based on the current sensor configuration.

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, and Sanjay Jha, "The Holes Problem in Wireless Sensor Networks: A survey," Proc. ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), vol. 9, no. 2, pp. 4-18, 2005.

[2] F. Aurenhammer, "Voronoi Diagrams – A Survey of fundamental Geometric Data Structure," ACM Computing Surveys, vol. 23, no. 3, pp. 345-405, 1991.

[3] Mihaela Cardei and Jie Wu, "Coverage in Wireless Sensor Networks," Handbook of Sensor Networks," M. Ilyas and I. Magboub (eds.), CRC Press, 2004.

[4] K. Chakrabarty, S. S. Iyengar, H. Qi and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," IEEE Trans. on Computers, vol. 51, pp. 1448-1453, 2002.

[5] O. Devillers, "Improved incremental randomized Delaunay triangulation," Proc. the Fourteenth Annual Symposium on Computational Geometry, pp. 106-115, ACM Press, 1998

[6] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," Proc. IEEE Wireless Communications and Networking Conference, pp. 1609-1614, 2003.

[7] D. Du and F. Hwang S. Fortune, "Voronoi diagrams and Delaunay triangulations," Euclidean Geometry and Computers, 1992.

[8] A. Howard, M.J. Mataric, and G.S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," Autonomous Robots, Special Issue on Intelligent Embedded Systems, 2002.

[9] A. Howard, M.J. Mataric, and G.S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," Proc. 6th International Symposium on Distributed Autonomous Robotic Systems, pp. 299-308, 2002.

[10] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis, "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea," Proc. the 3rd international conference on Embedded networked sensor systems (SenSys '05), pp. 64-75, ACM Press, 2005.

[11] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava, "Worst and Best-Case Coverage in Sensor Networks," IEEE Trans. on Mobile Computing, vol. 4, no. 1, 2005.

[12] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.Srivastava "Coverage Problems in Wireless Ad-Hoc Sensor Network," Proc. IEEE INFOCOM, pp. 1380-1387, 2001.

[13] S. Shakkottai, R. Srikant, N. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," Proc. IEEE INFOCOM, pp. 1073-1083, 2003.

[14] Guiling Wang, Guohong Cao, and Tom La Porta, "Movement-Assisted Sensor Deployment," Proc. IEEE INFOCOM, pp. 2469-2479, 2004.

[15] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks," Wireless Internet Conf. (WICON), 2005.

[16] Geoffrey Werner-Allen, Konrad Lorincz, Mario Ruiz, Omar Marcillo, Jeff Johnson, Jonathan Lees and Matt Welsh, "Deploying a Wireless Sensor Network on an Active Volcano," In press. To appear early 2006 in special Sensor Nets issue of IEEE Internet Computing.

[17] Ning Xu, "A Survey of Sensor Network Applications," University of Southern California, http://enl.usc.edu/~ningxu/papers/survey.pdf, 2003.

[18] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," Proc. IEEE INFOCOM, pp. 1293-1303, 2003.