Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks

You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng*
Department of Computer Science and Information Engineering
National Chiao Tung University, Hsin-Chu, 30010, Taiwan
Email:{wangyc, cchu, yctseng}@csie.nctu.edu.tw

Abstract

Sensor deployment is a critical issue since it reflects the cost and detection capability of a wireless sensor network. Although lots of work has addressed this issue, most of them assume that the sensing field is an open space and there is a special relationship between the communication range and sensing range of sensors. In this work, we consider the sensing field as an arbitrary-shaped region possibly with obstacles. Besides, we allow an arbitrary relationship between the communication range and sensing range, thus eliminating the constraints of existing results. Our approach is to partition the sensing field into smaller sub-regions based on the shape of the field, and then to deploy sensors in these sub-regions. Simulation results show that our method requires fewer sensors compared to existing results.

1 Introduction

Recently, wireless sensor networks have been studied intensively for applications such as monitoring physical environments. A wireless sensor network is composed of many tiny, low-power nodes that integrate sensing units, transceivers, and actuators with limited on-board processing and wireless communication capabilities [1]. These devices are deployed in a region of interest to gather information from the environment, which will be reported to a remote base station. Wireless sensor networks have been considered in many potential applications, such as surveillance, biological detection, and traffic, pollution, habitat, and civil infrastructure monitoring [2, 3, 6, 11, 13].

Sensor deployment is a critical issue since it reflects the cost and detection capability of a wireless sensor network. A good deployment should consider both *coverage* and *connectivity* [14, 18, 21]. Coverage requires that every location in the sensing field is monitored by at least one sensor. Connectivity requires that the network is not partitioned in terms of nodes' communication capability. Note that coverage is affected by sensors' sensitivity, while connectivity is influenced by sensors' communication ranges.

The *art gallery* problem has been studied extensively previously [7, 15, 17]. The problem asks how to use a minimum set of guards in a polygon such that every point of the polygon is watched by at least one guard. It is typically assumed that a guard can watch a point if line-of-sight exists. So the results cannot be directly applied to the sensor deployment problem since the sensing range of a sensor is normally finite. Besides, the art gallery problem does not address the communication issue between guards. Therefore, several methods have been proposed to solve the deployment problem for sensor networks. The work in [20] mainly discusses how to adjust sensors' locations to satisfy the coverage requirement in an open space, but without considering obstacles. The work in [12, 22] do consider sensing fields with obstacles when deploying sensors, but the results are limited to the special case when communication ranges are equal to sensing ranges. The work in [4, 5, 16] place sensors in a grid-like manner to satisfy coverage and connectivity. However, such approaches are not efficient in terms of the number of sensors being used. How to adaptively put sensors into the sleep mode to save energy while maintain full coverage of the sensing fields is proposed in [10, 19, 21]. The goal is different from our work, which assumes that we can choose the locations to deploy sensors. Also, such work normally assumes that the communication ranges of sensors are much larger than their sensing ranges.

In this work, we consider the sensing field as an arbitrary-shaped region possibly with obstacles. An obstacle can have any shape too. So the results may model an indoor environment. Also, we do not assume any relationship between sensing ranges and communication ranges, thus eliminating the constraints of existing deployment schemes.

*Y. C. Tseng's research is co-sponsored by the NSC Program for Promoting Academic Excellence of Universities under grant number 93-2752-E-007-001-PAE, by Computer and Communications Research Labs., ITRI, Taiwan, and by Intel Inc.

Our approach is to partition the sensing field into smaller sub-regions according to obstacles. Then sensors are deployed in each sub-region. Our simulations show that fewer sensors are required compared to existing results.

The rest of this paper is organized as follows. Section 2 formally defines the problem and reviews some related work. Sections 3 and 4 propose our sensor deployment algorithms. Simulation results are presented in Section 5. Conclusions are drawn in Section 6.

2 Preliminaries

2.1 Problem Definition

We are given a sensing field A in which sensors are to be deployed. Each sensor has a communication range r_c, within which it can transmit packets to other sensors, and a sensing range r_s, within which it can correctly monitor. We assume that all sensors have the same r_c and r_s. However, we make no assumption about the relationship between r_c and r_s. Our goal is to deploy sensors in A to ensure both *sensing coverage* (i.e., every point in A can be monitored) and *network connectivity* (i.e., no sensor gets disconnected) using as few sensors as possible.

The sensing field A is modeled by an arbitrary polygon on a 2D plane. Obstacles may exist inside A, which are also modeled as polygons. However, obstacles do not partition A (otherwise, maintaining network connectivity wouldn't be possible). For obstacles with arc or curve boundaries, we can approximate them by polygons. With the presence of obstacles, we define two sensors S_i and S_j to be *connected* if $|\overline{S_iS_j}| \leq r_c$ and $\overline{S_iS_j}$ does not intersect any obstacle or A's boundary; otherwise, they are *disconnected*. Fig. 1 shows two examples about the connectivity of two sensors. Obstacles may also reduce the coverage of a sensor. We assume that a point can be monitored by a sensor if it is within a distance of r_s and line-of-sight exists with the existence of obstacles. Fig. 2 shows two examples. Note that the above definitions assume that sensors need line-of-sight to sense/communicate. Although the assumption may be conservative, it does guarantee better coverage of the field and better connectivity among sensors. If this assumption is removed, our results can even be simplified. Also note that the sensing field A may already contain some sensors, which can be easily treated as a special case of obstacles.

We conclude the discussion by a sensor deployment example in an office environment as shown in Fig. 3. Note that we assume $r_c = r_s$ in this example.

2.2 Related Work

Some work assumes mobile sensors. The work [22] proposes a *virtual force* algorithm to enhance coverage after an

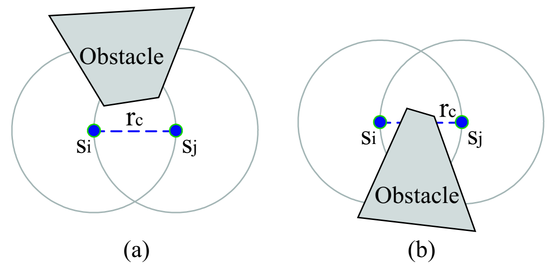


Figure 1. (a) S_i and S_j are connected, and (b) the obstacle disconnects S_i and S_j.

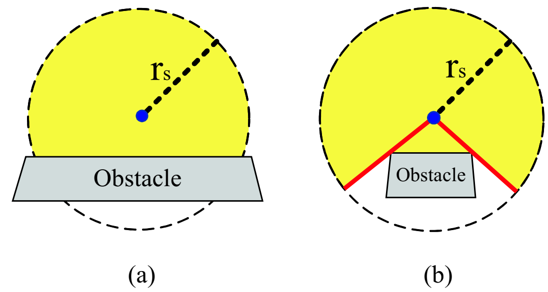


Figure 2. The coverage of a sensor blocked by obstacles (shaded areas are covered).

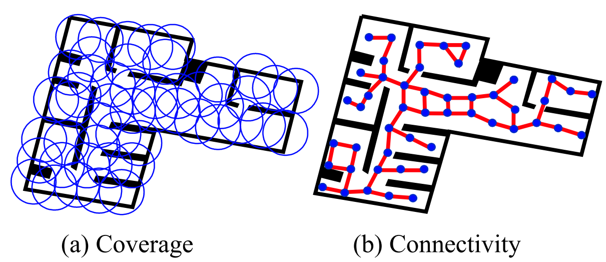


Figure 3. A sensor deployment example in an office environment.

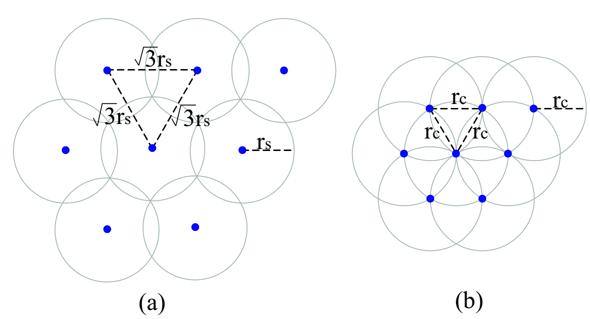
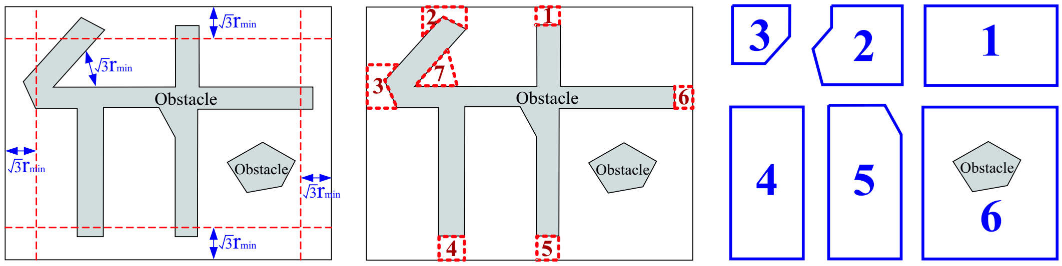


Figure 4. Two intuitive deployment solutions: (a) considering coverage property first and (b) considering connectivity property first.



(a) A sensing field with obstacles (b) Small regions (c) Large regions

Figure 5. Partitioning a sensing field: (a) the sensing field, (b) small regions, and (c) large regions.

initial random placement of sensors. Sensors will be moved by the attractive or repulsive forces of neighboring sensors and obstacles. In [20], Voronoi diagrams are used to discover coverage holes after the initial deployment. Sensors are then moved from densely deployed areas to these holes.

The work in [4, 5, 16] place sensors in a grid-like manner to satisfy coverage and connectivity. It is clear that a hexagon-like placement saves more sensors. So this kind of deployment is not efficient, especially when there are arbitrary relationships between r_c and r_s. Besides, obstacles may destroy the regularity of grids. In [12], it is suggested to deploy sensors along the x-axis by a distance of r_c and then along the y-axis by a distance of $(1 + \frac{\sqrt{3}}{2})r_c$. However, lots of sensors are needed to satisfy connectivity when $r_c \geq \sqrt{3}r_s$. The work [21] indicates that when $r_c \geq 2r_s$, full coverage will guarantee connectivity. Besides, to satisfy full coverage, the distance between adjacent sensors should be $\sqrt{3}r_s$. The result is very limited since only special relationship of r_s and r_c is considered. Also, obstacles are not considered.

Sensor deployment is also addressed in the field of robotics [9, 8]. With robots, sensors can be deployed one by one. The information gathered by deployed sensors can be used to determine the location of the next sensor.

2.3 Two Naive Deployment Algorithms

The sensor deployment problem does pose much challenge. Below, we make some observations based on two extreme solutions. The first one tries to satisfy the coverage property first. In this scheme, to keep a minimal number of sensors, we have to minimize the overlapping coverage as much as possible. The result would be as shown in Fig. 4 (a), where neighboring sensors are evenly separated by a distance of $\sqrt{3}r_s$. This scheme is very efficient when $r_c \geq \sqrt{3}r_s$ since connectivity is automatically guaranteed. However, when $r_c < \sqrt{3}r_s$, extra sensors have to be added to maintain connectivity. Inefficiency may be incurred because all sensing field has been covered and these newly added sensors will not make any contribution to coverage.

The second solution is to satisfy the connectivity property first. This will result in a deployment as shown in Fig. 4 (b), where neighboring sensors are evenly separated by r_c. This scheme will be very efficient when $r_c \leq \sqrt{3}r_s$ because coverage is automatically guaranteed. However, when $r_c > \sqrt{3}r_s$, extra sensors have to be added to maintain coverage. Inefficiency may be incurred because the overlapping coverage could be large.

3 Deployment Algorithms

Given a sensing field A, our goal is to deploy as few sensors as possible to maintain both coverage and connectivity. We first partition A into a number of regions, each being a polygon. Regions are classified as *large* and *small*. We define a *small region* as a belt-like area between obstacles or A's boundary, and its width is not larger than $\sqrt{3}r_{min}$, where $r_{min} = min(r_s, r_c)$. Excluding small regions, the other regions are *large regions*. Fig. 5 gives an example to partition a sensing field. There are seven small regions and six large regions. Note that a region may still exist obstacles, e.g., region 6. How to partition a sensing field will be discussed in Section 4.

Below, we discuss how to deploy sensors in a single region. Note that in our schemes, extra sensors will be deployed on boundaries of regions, so connectivity between different regions are automatically guaranteed.

3.1 Deploying Sensors in Small Regions

For a small region, we can find its bisector and then deploy a sequence of sensors along the bisector to satisfy both coverage and connectivity. How to find a bisector of a region can be achieved by doing a triangulation on that region, as shown in Fig. 6. A bisector can be formed from connecting the midpoints of all dotted lines. Note that if the end of a small region forms a corner (e.g., the case of Fig. 6(b)), then the corner is also considered a midpoint. After finding a bisector, we can deploy a sequence of sensors by a distance of r_{min} along each line segment of the bisector to ensure coverage and connectivity of that region, as shown in Fig. 6. Note that we always add an extra sensor at the end of the bisector for ensuring connectivity to neighboring regions.

3.2 Deploying Sensors in Large Regions

A region that cannot be simply covered by a sequence of sensors as above is treated as a large region. Multiple rows of sensors will be needed. Below, we first consider a simple large region without boundaries and obstacles. Then we extend our result to an environment with boundaries and obstacles.

3.2.1 Simple Large Regions

Given a 2D plane without boundaries and obstacles, we will deploy sensors row by row. The basic idea is to form a row of sensors that is connected. Adjacent rows should guarantee continuous coverage of the area. Finally, we will add some sensors between adjacent rows, if necessary, to maintain connectivity. Based on the relationship between r_s and r_c, we separate the discussion into two cases.

Case 1: $r_c \leq \sqrt{3}r_s$. In this case, sensors on each row are separated by a distance of r_c. So the connectivity of sensors in each row is already guaranteed. Since $r_c < \sqrt{3}r_s$, each row of sensors can cover a belt-like area with a width of $2 \times \sqrt{r_s^2 - \frac{r_c^2}{4}}$. Adjacent rows will be separated by a distance of $r_s + \sqrt{r_s^2 - \frac{r_c^2}{4}}$ and shifted by a distance of $\frac{r_c}{2}$. With such an arrangement, the coverage of the whole area is guaranteed. Fig. 7(a)–(c) show three possible cases. Note that in the case of $r_c < \sqrt{3}r_s$, the distance between two adjacent rows is larger than r_c, so we need to add a column of sensors between two adjacent rows, each separated by a distance no larger than r_c, to connect them.

Case 2: $r_c > \sqrt{3}r_s$. In this case, the previous approach will waste a lot of sensors because the small r_s requires two rows to be very close. So when $r_c > \sqrt{3}r_s$, we propose to deploy sensors in a typical hexagon manner such that adjacent sensors are regularly separated by a distance of $\sqrt{3}r_s$. Both coverage and connectivity properties are satisfied.

3.2.2 Large Regions with Boundaries and Obstacles

Next, we modify the above solution for deploying sensors in a region with boundaries and obstacles. Observe that in our solution, sensors are deployed in regular patterns. Thus, the above solution can be transformed into an incremental approach where sensors are added into the field one by one. In Table 1, we summarize the coordinates of a sensors's six neighbors. Thus, we can first place a sensor in any location of the region, from which the six locations that can potentially be deployed with sensors are determined.

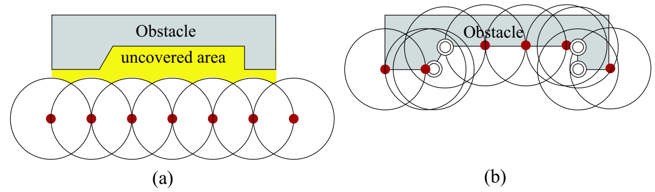


Figure 8. (a) uncovered area around an obstacle, and (b) extra sensors along the boundary to cover the uncovered area.

These locations are inserted into a queue Q. We then enter a loop in which each time an entry (x, y) is dequeued from Q. If (x, y) is not inside any obstacle and not outside of the region, a sensor will be placed in (x, y). Also, the six neighboring locations are calculated according to Table 1 and inserted into Q if they have not be deployed with sensors. This process is repeated until Q becomes empty.

The above approach may leave three problems unsolved. First, some areas near the boundaries or obstacles may be left uncovered. Second, as mentioned before, when $r_c < \sqrt{3}r_s$, we need to add extra sensors between adjacent rows to maintain connectivity. Third, connectivity to neighboring regions needs to be maintained. These problems can be easily solved by sequentially placing sensors along the boundaries of the region and obstacles. Fig. 8 gives an example (we assume that $r_s = r_c$). Note that since obstacles may disconnect adjacent sensors, extra sensors may need to be placed at corners of obstacles (shown by double circles in Fig. 8(b)). There are two cases for the distance between adjacent sensors:

- When $r_c \leq \sqrt{3}r_s$, since the maximum width of the uncovered area does not exceed r_c, sensors should be separated by r_c.
- When $r_c > \sqrt{3}r_s$, since the maximum width of the uncovered area does not exceed $\sqrt{3}r_s$, sensors should be separated by $\sqrt{3}r_s$. Since $r_c > \sqrt{3}r_s$, the connectivity between these extra-added sensors and the regularly deployed sensors are guaranteed.

4 Partitioning a Sensing Field into Small and Large Regions

Section 3 does not explain how to partition the sensing field A into small and large regions. Below, we show how to identify small regions. After excluding small regions, the remaining regions are considered large.

To identify small regions, we first expand the perimeters of obstacles outwardly and A's boundaries inwardly by a distance of $\sqrt{3}r_{min}$. Such an expansion may cause overlapping with the original obstacles and A's boundary. For

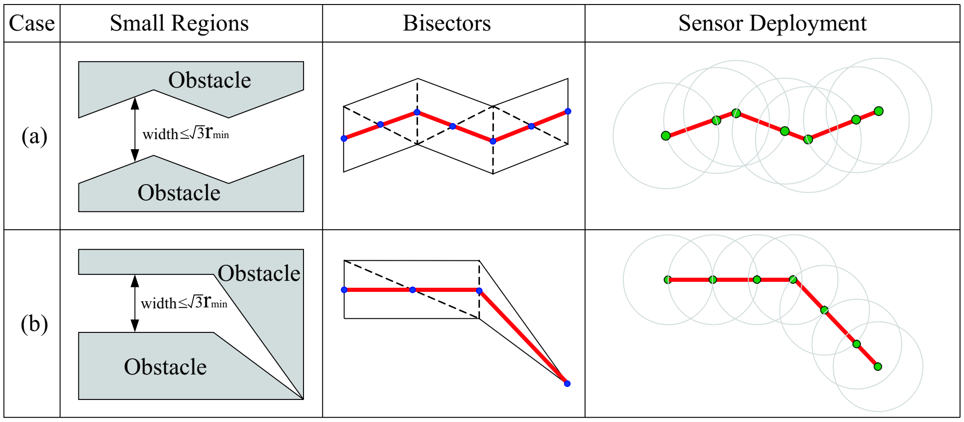


Figure 6. Two examples to find bisectors of small regions and the corresponding sensor deployments.

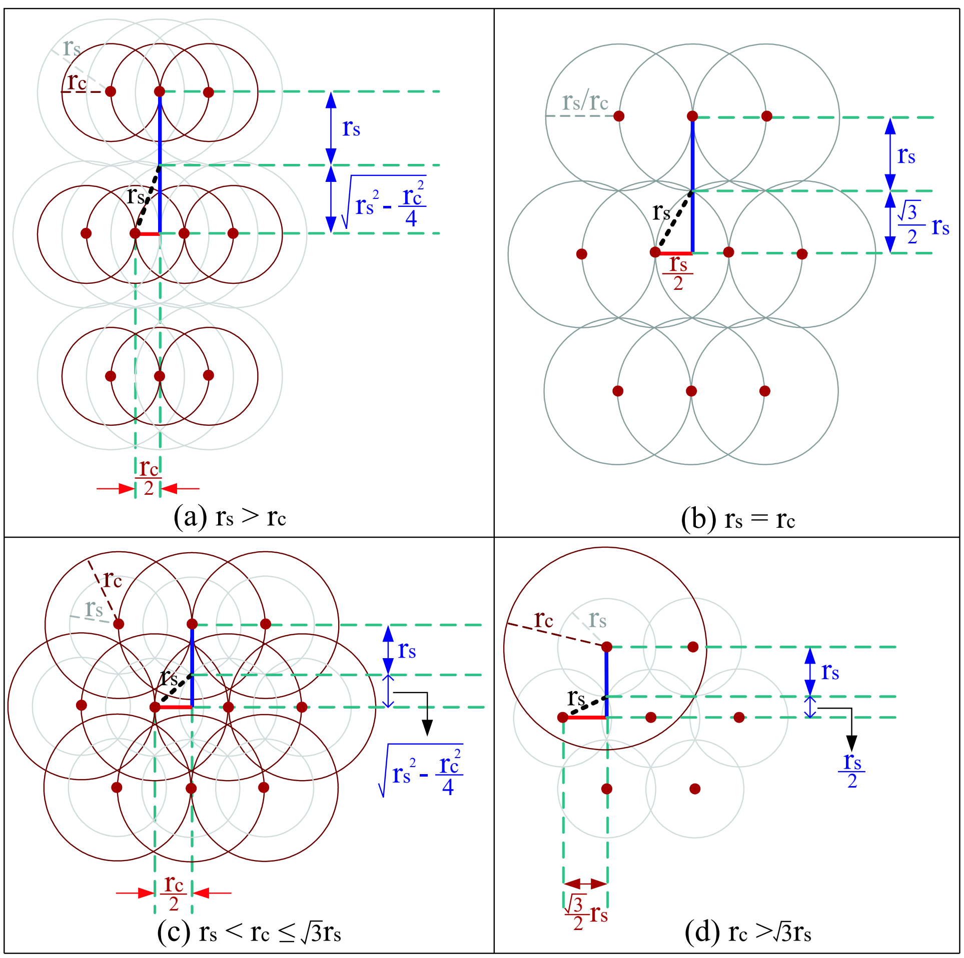


Figure 7. Deploying sensors in simple large regions.

Table 1. Coordinates of the six neighbors of a sensor in location (x, y).

Neighbor	$r_c \le \sqrt{3}r_s$	$r_c > \sqrt{3}r_s$
N_1	$(x + r_c, y)$	$(x + \sqrt{3}r_s, y)$
N_2	$(x + \frac{r_c}{2}, y - \sqrt{r_s^2 - \frac{r_c^2}{4}} - r_s)$	$(x + \frac{\sqrt{3}r_s}{2}, y - \frac{3r_s}{2})$
N_3	$(x - \frac{r_c}{2}, y - \sqrt{r_s^2 - \frac{r_c^2}{4}} - r_s)$	$(x - \frac{\sqrt{3}r_s}{2}, y - \frac{3r_s}{2})$
N_4	$(x - r_c, y)$	$(x - \sqrt{3}r_s, y)$
N_5	$(x - \frac{r_c}{2}, y + \sqrt{r_s^2 - \frac{r_c^2}{4}} + r_s)$	$(x - \frac{\sqrt{3}r_s}{2}, y + \frac{3r_s}{2})$
N_6	$(x + \frac{r_c}{2}, y + \sqrt{r_s^2 - \frac{r_c^2}{4}} + r_s)$	$(x + \frac{\sqrt{3}r_s}{2}, y + \frac{3r_s}{2})$

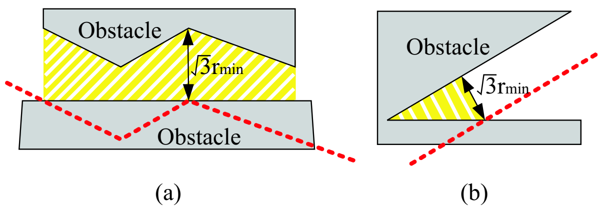


Figure 9. Two examples to find small regions. The dotted lines are expansions of obstacles.

those parts with overlapping, we can take a projection back to the original perimeters to obtain some small regions. Taking Fig. 5(a) as an example, the dotted lines are expansion of A's boundaries. For these overlaps, we can take a projection to obtain small regions, as numbered from 1 to 6 in Fig. 5(b). Fig. 9 shows two examples of the expansions of obstacles. Note that the above expansions may result in multiple different small regions in the same place. In this case, we can select the largest one as a small region.

5 Simulation Results

In this section, we present some experimental results to verify the effectiveness of the proposed sensor deployment algorithm. We design six kinds of sensing fields, as shown in Fig. 10. We consider four cases: $(r_s, r_c) = (7, 5)$, $(5, 5)$, $(3.5, 5)$, and $(2, 5)$ to reflect the relationships of $r_s > r_c$, $r_s = r_c$, $r_s < r_c \le \sqrt{3}r_s$, and $\sqrt{3}r_s < r_c$, respectively. We mainly compare our algorithm and two deployment methods discussed in Section 2.3 (namely coverage-first and connectivity-first methods). The comparison metric is the average number of sensors being used (for each field, we place the first sensor at different location, and average the results of all deployments).

Fig. 11 compares the number of sensors being used when $r_c \le \sqrt{3}r_s$ in different sensing fields. The connectivity-first method is dominated by the value of r_c, so the number of sensors is fixed when $r_c \le \sqrt{3}r_s$. Thus, when $r_s \ge r_c$, this method uses the most sensors because the overlapping in coverage is very large. On the contrary, when $r_s < r_c \le \sqrt{3}r_s$, the coverage-first method uses the most sensors, because it needs many extra sensors to maintain connectivity between neighboring sensors. The proposed method uses the least sensors because it can adjust the distance between two adjacent rows according to the relationship of r_s and r_c.

Fig. 12 makes a similar comparison when $r_c > \sqrt{3}r_s$. Our algorithm still uses the least sensors in all cases. Note that when $r_c > \sqrt{3}r_s$, our algorithm works the same as the coverage-first method in each individual region, so we omit its performance in Fig. 12.

6 Conclusions

In this work, we have proposed a systematical solution for sensor deployment. The sensing field is modeled as an arbitrary polygon possibly with obstacles. Thus, the result can be used in an indoor environment. The result can be applied to sensors with arbitrary relationships of communication ranges and sensing ranges. Fewer sensors are required to ensure fully coverage of the sensing field and connectivity of the network as compared to other methods. Note that in this work we assume that sensors have predictable communication range r_c and sensing range r_s. This may result in fragile networks when the terrain factor is concerned. To resolve this problem, we can substitute r_c and r_s by r'_c and r'_s which are slightly smaller than r_c and r_s, respectively. This should result in a stronger network. Also, in our solution in Section 3.2.1, we can add more columns of sensors among adjacent rows to improve the reliability of the network.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

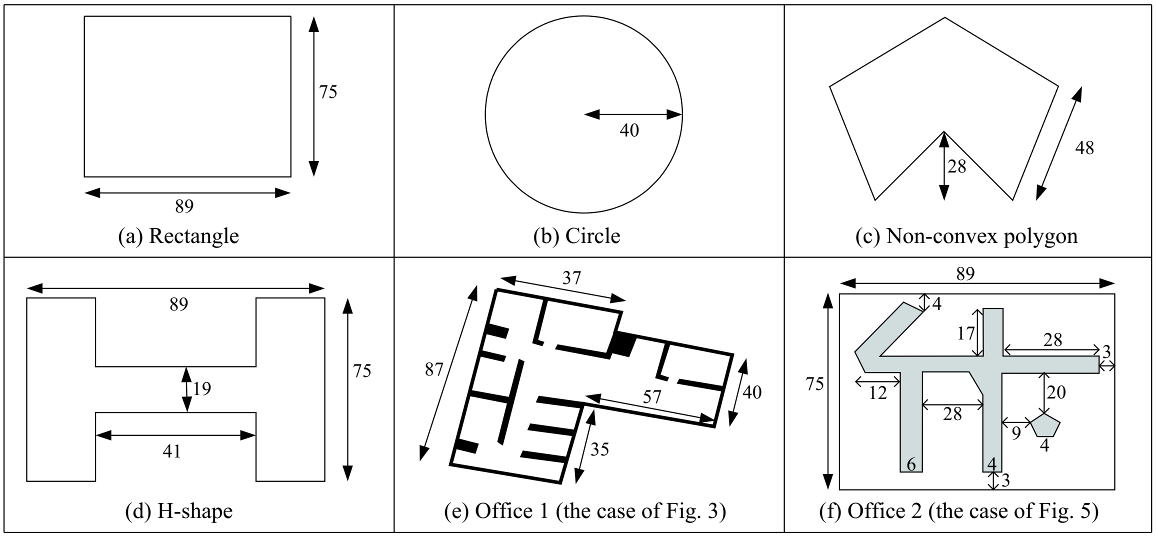


Figure 10. Sensing fields used in the simulations.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[3] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38–45, 2004.

[4] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy for target detection. In *ACM International Workshop on Wireless Sensor Networks and Applications*, pages 42–48, 2002.

[5] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *International Conference on Information Fusion*, pages 1581–1587, 2002.

[6] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, 1999.

[7] F. Hoffmann, M. Kaufmann, and K. Kriegel. The art gallery theorem for polygons with holes. In *IEEE Symposium on Foundations of Computer Science*, pages 39–48, 1991.

[8] A. Howard, M. J. Matarić, and G. S. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Auton. Robots*, 13(2):113–126, 2002.

[9] A. Howard, M. J. Matarić, and G. S. Sukhatme. Mobile sensor networks deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *International Symposium on Distributed Autonomous Robotics Systems*, 2002.

[10] C. F. Huang and Y. C. Tseng. The coverage problem in a wireless sensor network. In *ACM International Workshop on Wireless Sensor Networks and Applications*, pages 115–121, 2003.

[11] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for "Smart Dust". In *ACM/IEEE International Conference on Mobile Computing and Networking*, pages 271–278, 1999.

[12] K. Kar and S. Banerjee. Node placement for connected coverage in sensor networks. In *Proceedings of WiOpt*, 2003.

[13] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[14] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, pages 1380–1387, 2001.

[15] I. B. Sachs and D. L. Souvaine. An efficient algorithm for guard placement in polygons with holes. *Discrete & Computational Geometry*, 13:77–109, 1995.

[16] S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: coverage, connectivity and diameter. In *INFOCOM*, pages 1073–1083, 2003.

[17] T. C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[18] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *ACM International Workshop on Wireless Sensor Networks and Applications*, pages 32–41, 2002.

[19] D. Tian and N. D. Georganas. A node scheduling scheme for energy conservation in large wireless sensor networks. In *Wireless Communication and Mobile Computing (WCNC)*, pages 271–290, 2003.

[20] G. Wang, G. Cao, and T. L. Porta. Movement-assisted sensor deployment. In *INFOCOM*, pages 2469–2479, 2004.

[21] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. In *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2004.

[22] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *INFOCOM*, pages 1293–1303, 2003.

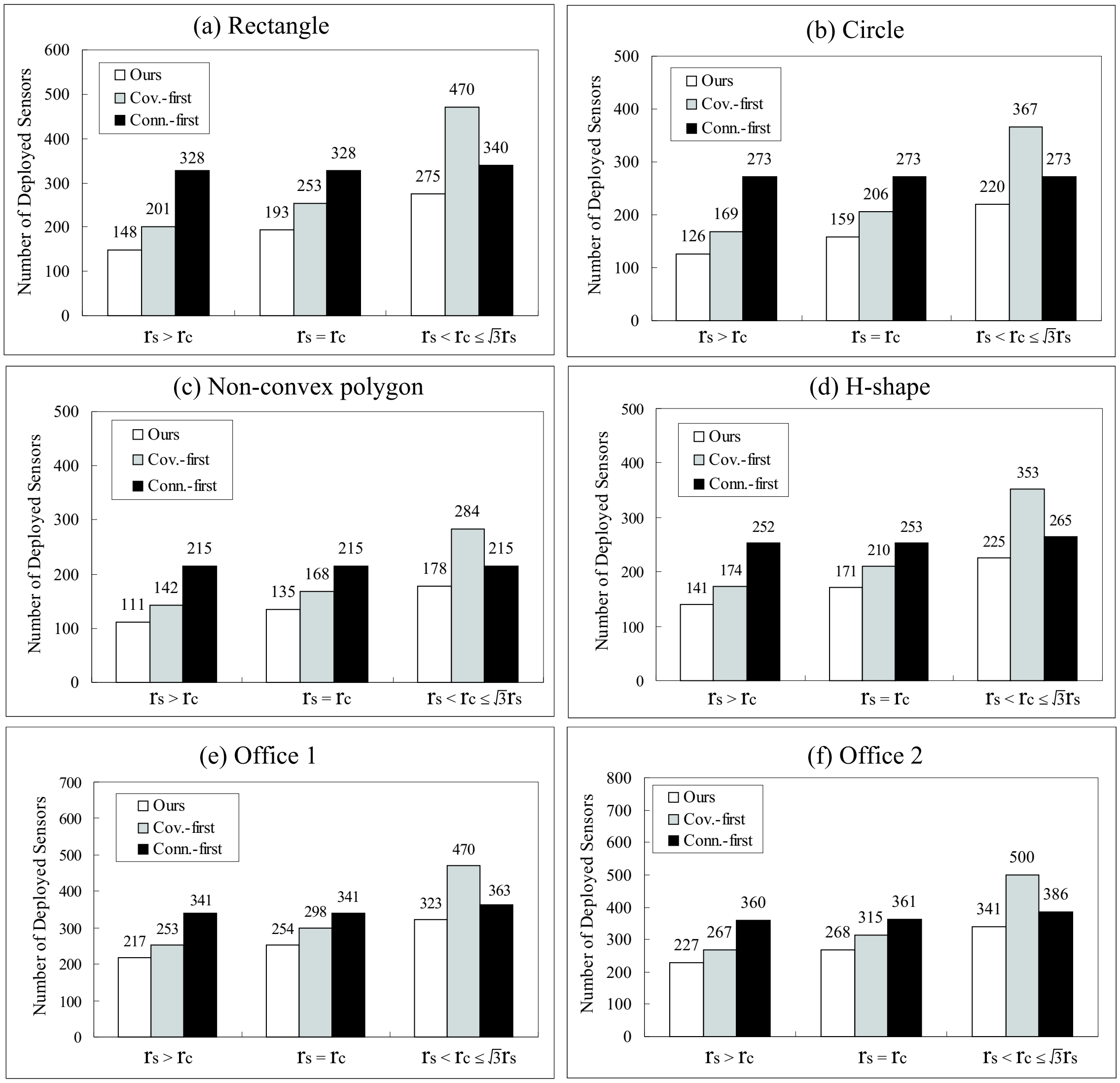


Figure 11. Average number of sensors used when $r_c \leq \sqrt{3}r_s$ under different shapes of sensing fields.

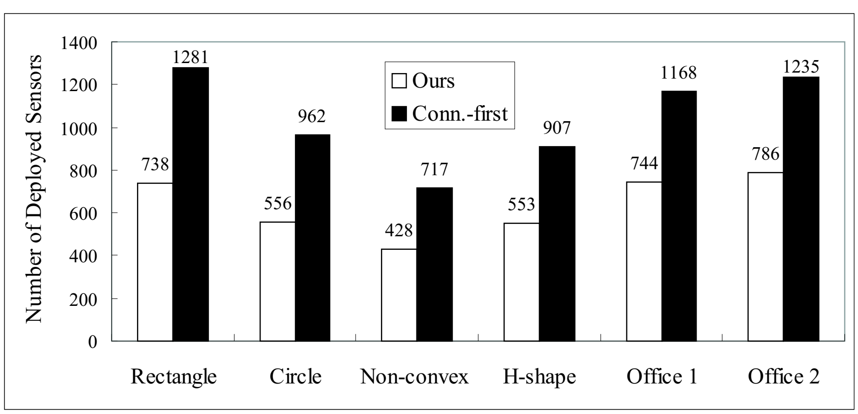


Figure 12. Average number of sensors used when $r_c > \sqrt{3}r_s$ under different shapes of sensing fields.