無線感測網路上有效確保完全覆蓋與通訊之感測器部署演算法

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

Student: Chun-Chi Hu

Advisor: Prof. Yu-Chee Tseng

國立交通大學
資訊工程學系
碩士論文

A Dissertation Submitted to
the Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao-Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science and Information Engineering

Hsinchu, Taiwan

April 2005

無線感測網路上有效確保完全覆蓋與通訊之感測器部署演算法

學生：胡君琪　　　　　　　指導教授：曾煜棋 教授

國立交通大學資訊工程學系(研究所)碩士班

摘　　　要

在無線感測網路的各種研究中，感測器的布署方式是一項重要的議題，部署的方式將會反應整個網路的效能與偵測能力。雖然目前已有許多相關研究曾針對於此議題提出討論，但大多數的討論都只針對於在開放的感測區域中部署或使用特定比例的感測與通訊範圍的感測器作為部署工具。在本論文中，我們允許感測器部署的環境是為任意形狀的區域，並且於其中可能存在有任意形狀的障礙物。除此之外，我們也允許感測器的感測範圍與通訊範圍之間可為任意比例。我們提出的部署方式，首先將分析部署環境，將部署環境分為大區域與小區域兩種，並根據兩種區域的特性提出不同的部署方式。模擬環境部署的結果，可以顯示我們所提出的部署方式可比現有的部署方式節省較多感測器的使用。

關鍵字：連接，覆蓋，網路部署，感測網路，拓樸控制，無線網路。

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

Student: Chun-Chi Hu Advisors: Prof. Yu-Chee Tseng

Institute of Computer Science and Information Engineering
National Chiao-Tung University

ABSTRACT

Sensor deployment is an important issue since it reflects the cost and detection capability of a wireless sensor network. Although a lot of work has addressed this issue, most of them assume that the sensing field is an open space and that there exists 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 sensing range and communication range of sensors, thus eliminating the constraints of existing results. Our approach is to partition the sensing field into small subregions according to the shape of the field. Simulation results are presented, which do show that our result requires fewer sensors compared to existing results.

Keywords: connectivity, coverage, network deployment, sensor network, topology control, wireless networks.

Acknowledgments

My advisor, prof. Yu-Chee Tseng, is the first one I would like to express my gratitude to. With the wonderful research conditions he provided and his attentive instructions, I came to discover the pleasure of research. I am also grateful to my senior, You-Chiun Wang. Without his help and suggestions, I would not be able to have this thesis done. Finally, I would like to thank all HSCC members for their generous advices. Discussing with them benefited me in many ways.

ChunChi at CSIE, NCTU.

Contents

List of Figures

List of Tables

Chapter 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 [2]. 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 [8, 13, 15, 5, 3].

Sensor deployment is an important issue since it reflects the cost and detection capability of a wireless sensor network. A good deployment should take both *coverage* and *connectivity* properties into account [20, 23, 24, 16]. Coverage requires that every location in the sensing field can be 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.

There is a close resemblance between the sensor-deployment problem and the traditional art gallery problem [10, 19, 17]. The art gallery 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. However, it is typically assumed that a guard can watch a point as long as line-of-sight exists, so the results cannot be directly applied to the sensor deployment problem because 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 [22] mainly discusses how to adjust sensors' locations to satisfy the coverage requirement in an open space, but without considering obstacles. How to adaptively put sensors into the sleep mode to save energy while maintain full coverage of the sensing fields is proposed in [24, 12, 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 transmission ranges of sensors are much larger than their sensing ranges. The work in [26, 14] do consider sensing fields with obstacles when deploying sensors, but the results are only limited to the special case when communication ranges are equal to sensing ranges. The work in [18, 7, 6] 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.

In this work, we consider the sensing field as an arbitrary-shaped region with one or multiple obstacles. An obstacle can also have any shape. So the results can be model as an indoor environment. Also, we do not assume any relationship between sensing ranges and communication ranges, thus eliminating the constraints of existing deployment schemes. Our approach is to partition the sensing field into small 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 thesis 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.

Chapter 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 distance r_s, within which it can correctly monitor. We assume that all sensors have the same r_c and r_s. The coverage area of sensing distance and communication range of each sensor are assumed to be ideal circular shapes. 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* (in the sense that no point in A is unmonitored) and *network connectivity* (in the sense that 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 by polygons of arbitrary shapes. 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 the line segment $\overline{S_iS_j}$ does not intersect any obstacle or boundary of A; otherwise, they are *disconnected*. Fig. 2.1 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. 2.2 shows two examples. Note that the above definitions assume that sensors need line-of-sight to sense/communicate. Although this assumption is somewhat too

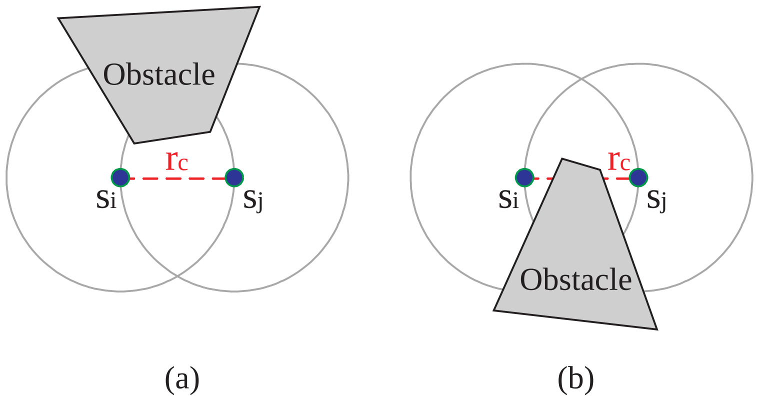


Figure 2.1: (a) the obstacle does not disconnect S_i and S_j, and (b) the obstacle disconnects S_i and S_j.

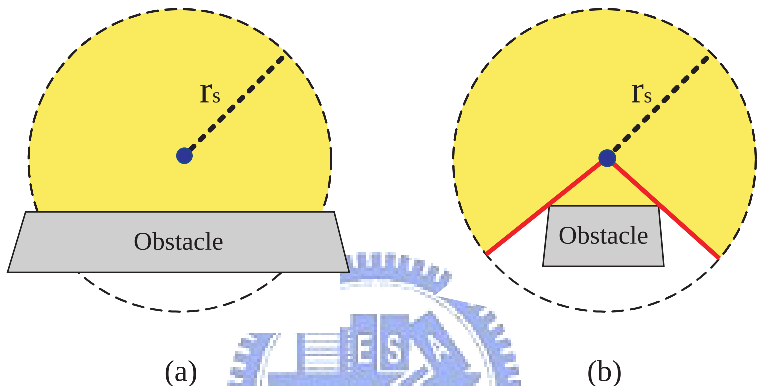


Figure 2.2: The coverage of a sensor blocked by obstacles (shaded areas are covered).

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. 2.3. Note that we assume $r_c = r_s$ in this example.

2.2 Related Work

The work in [18, 7, 6] 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 exists arbitrary relationship

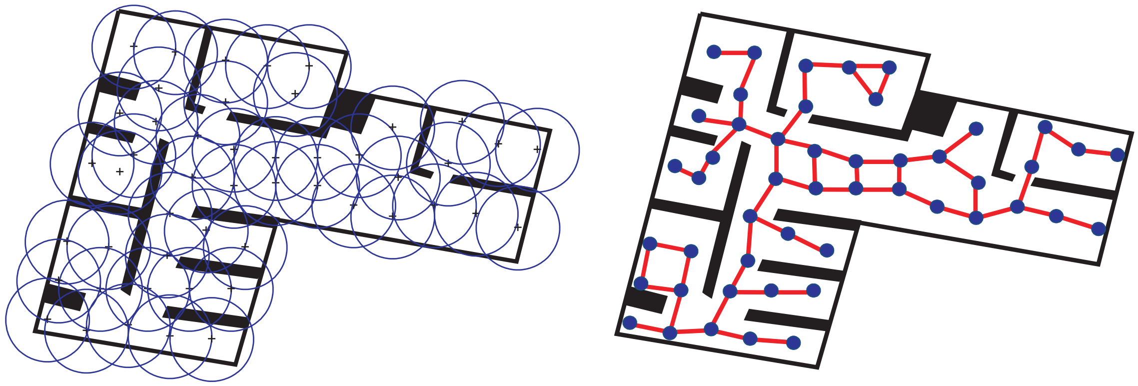
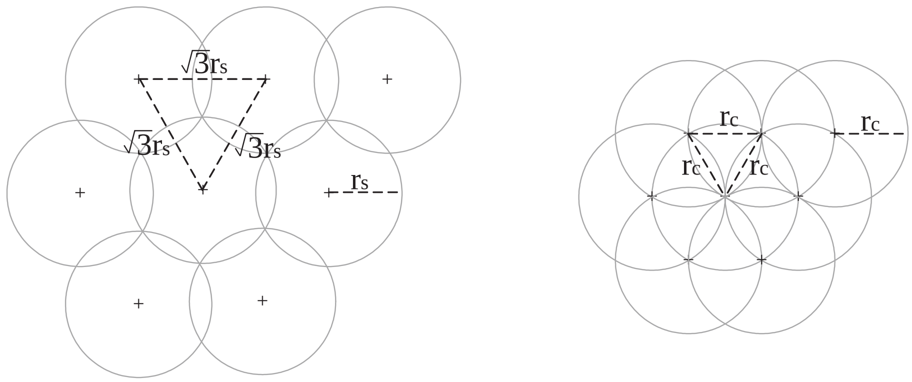


Figure 2.3: A sensor deployment example in an office environment: (a) coverage and (b) connectivity.

between communication ranges and sensing ranges. Besides, obstacles may destroy the regularity of grids. In [14], it is suggested to deploy sensors along the x-axis by the communication distance and then along the y-axis by the sensing distance. However, a lot of sensors is needed to satisfy connectivity when $r_c \geq \sqrt{3}r_s$. The work in [24] suggests that when $r_c \geq 2r_s$, full coverage will also guarantee connectivity. Besides, to satisfy full coverage, the distance between adjacent sensors should be $\sqrt{3}r_s$. Again, the result is very limited because 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 [11, 1]. 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. However, if the number of deployed sensors is too large or the deployment field is hostile, it is undesirable to deploy sensors one by one. Therefore, some works suggest to perform deployment by mobile sensors. The work in [25] adds new mobile sensors into an existing sensor network to enhance network coverage and connectivity. However, adding new sensors is usually a difficult job, especially when the deployed environment is hostile. The work in [4] fulfills network connectivity by moving some of nodes to new locations for the fault-tolerant purpose, but the coverage problem is ignored since it leaves some regions uncovered after moving these nodes. In [22], the Voronoi diagram is used to discover coverage holes after initial deployment. Sensors are then moved from densely deployed areas to these holes. The work in [9] suggests to use repulsive forces or the Voronoi diagram to decide the positions that sensors have to move.



(a) (b)

Figure 2.4: Two intuitive deployment solutions: (a) considering coverage property first and (b) considering connectivity property first.

However, both works ([22, 9]) consider an open-space environment. When there are obstacles inside the deployed field, both methods may not work. The work [26] proposes a *virtual force* concept to enhance coverage after an initial random placement of sensors. Sensors will be moved by the attractive or repulsive forces of neighboring sensors and obstacles. However, it cannot ensure full coverage of the deployed network since it does not consider the relationship between communication range and sensing range.

2.3 Some Observations

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, in order 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. 2.4 (a), where neighboring sensors are evenly separated by a distance of $\sqrt{3}r_s$. This scheme will be very efficient when $r_c \geq \sqrt{3}r_s$, because 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. 2.4 (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.

Chapter 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 whose 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. 3.1 gives an example to partition a sensing field. There are seven small regions and six large regions. Note that there may still exist obstacles in a region, e.g., region 6. How to partition a sensing field is 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

We define a small region as a belt-like area with a width no larger than $\sqrt{3}r_{min}$. We can then find a bisector of the region and deploy a row of sensors along the bisector to satisfy coverage and connectivity. Finding a bisector of a region is not a difficult job if we model the region by a polygon. For example, in Fig. 3.2, we first do a triangulation on each region. 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. 3.2(b)), then the corner is also considered a midpoint. After finding a bisector, we deploy a sequence of sensors along each line segment of the bisector with the interval distance of r_{min}, as shown in Fig. 3.2. Note that we

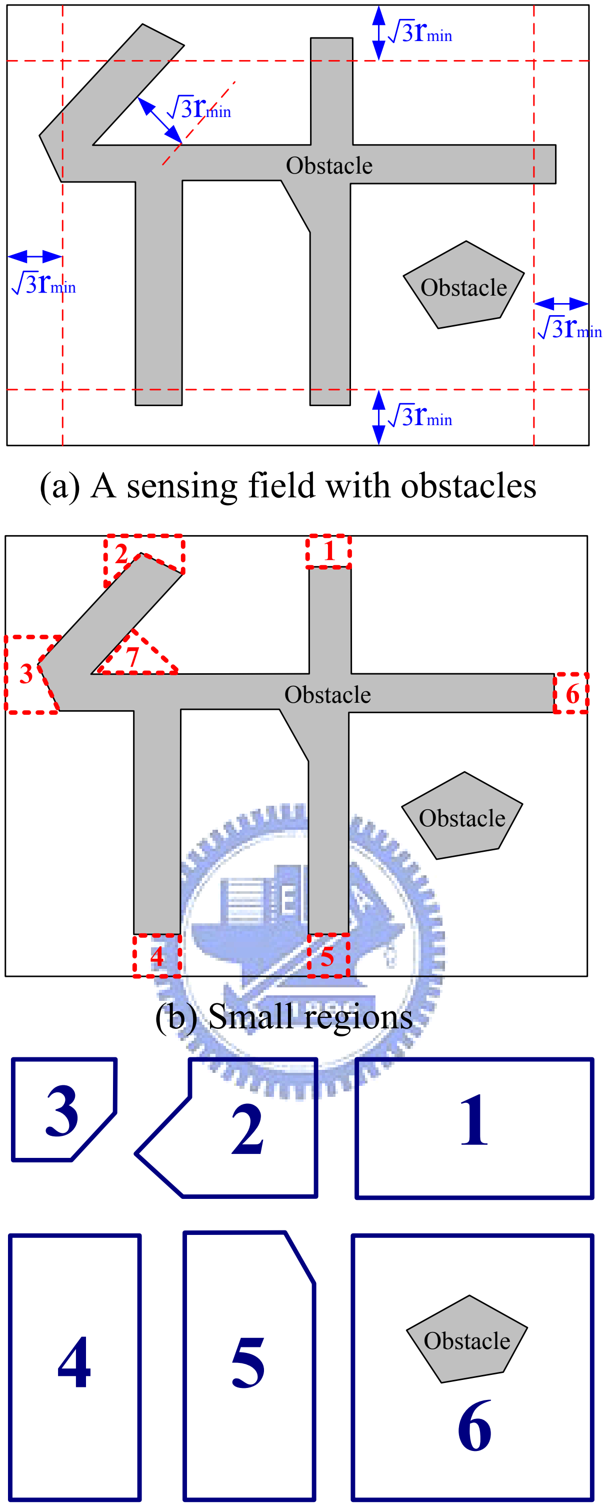


Figure 3.1: Partitioning a sensing field: (a) the sensing field, (b) small regions, and (c) large regions.

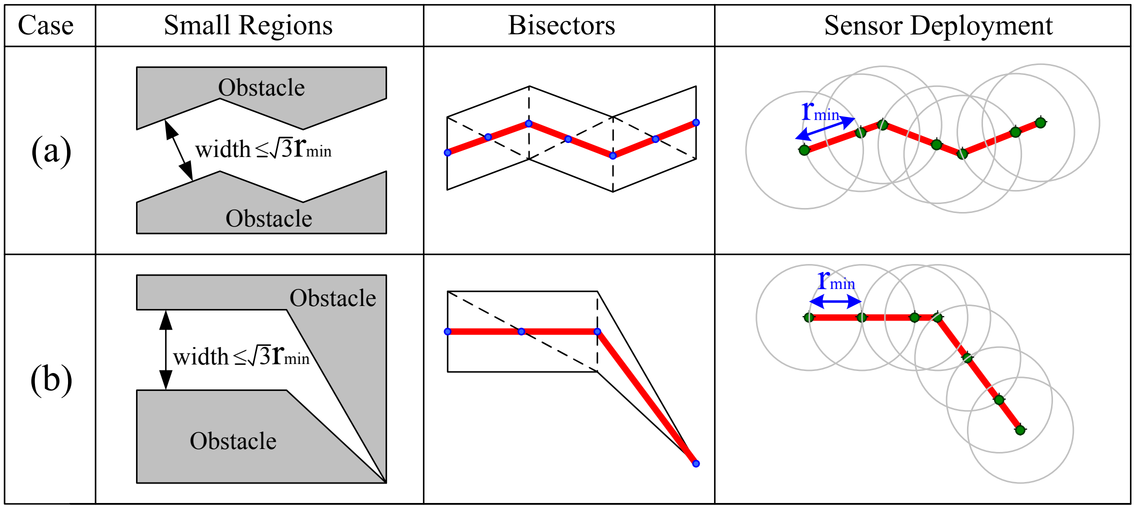


Figure 3.2: Two examples to find bisectors of small regions and the corresponding sensor deployments.

always add an extra sensor at the end of the bisector for ensuring connectivity to neighboring regions. The following lemma shows that our deployment method can guarantee coverage and connectivity of a small region.

Lemma 1 *By deploying sensors along the bisector with the distance of r_{min}, it is guaranteed to satisfy coverage and connectivity properties in a small region.*

Proof.

Since $r_{min} = min(r_s, r_c)$, it is clear that the deployed sensors satisfy the connectivity property. We then prove that such deployment can fully cover a small region whose width is no larger than $\sqrt{3}r_{min}$.

Case 1: *$r_s \geq r_c$. In this case, adjacent sensors are separated by the distance of r_c, so the width of the belt-like region that sensors in a line can cover is*

$$2 \times \sqrt{r_s^2 - \frac{r_c^2}{4}} \geq 2 \times \sqrt{r_c^2 - \frac{r_c^2}{4}} = \sqrt{3}r_c = \sqrt{3}r_{min}.$$

which is certainly larger than or equal to the width of a small region.

Case 2: *$r_s < r_c$. In this case, adjacent sensors are separated by the distance of r_s, so the width of the belt-like region that sensors in a line can cover will be exactly $\sqrt{3}r_s = \sqrt{3}r_{min}$.*

Therefore, in both cases, such deployment method can guarantee fully coverage of a small region. □

3.2 Deploying Sensors in Large Regions

A region that cannot be simply covered by a row 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 $r_c \leq \sqrt{3}r_s$, 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. 3.3(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. Two adjacent sensors in each row will have much uncovered region. 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 3.1, we summarize the

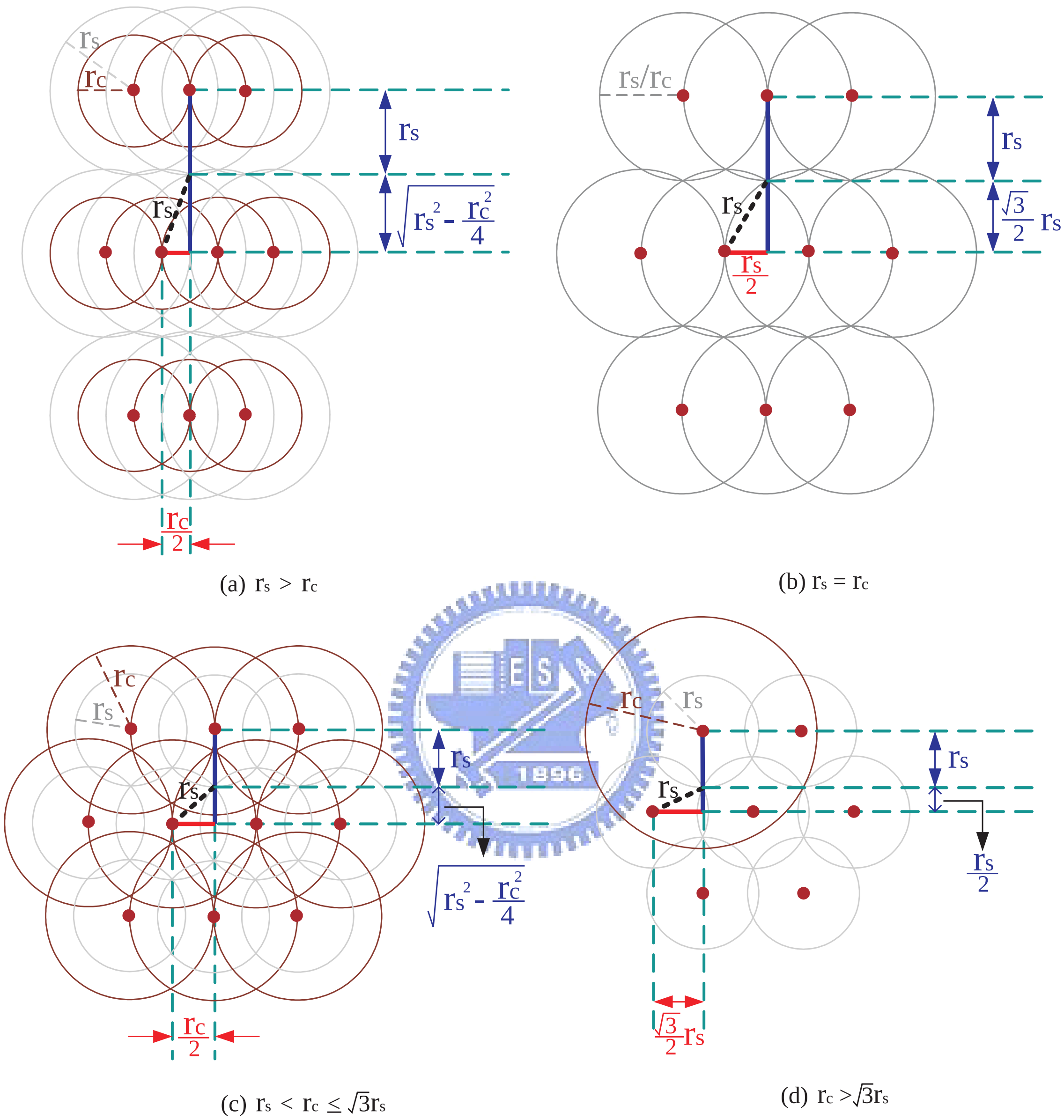


Figure 3.3: Deploying sensors in simple large regions: (a) $r_s > r_c$ (b) $r_s = r_c$ (c) $r_s < r_c \leq \sqrt{3}r_s$, and (d) $r_c > \sqrt{3}r_s$.

Table 3.1: Coordinates of the six neighbors of a sensor in location (x, y).

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

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. 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 3.1 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. 3.4 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. 3.4(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.

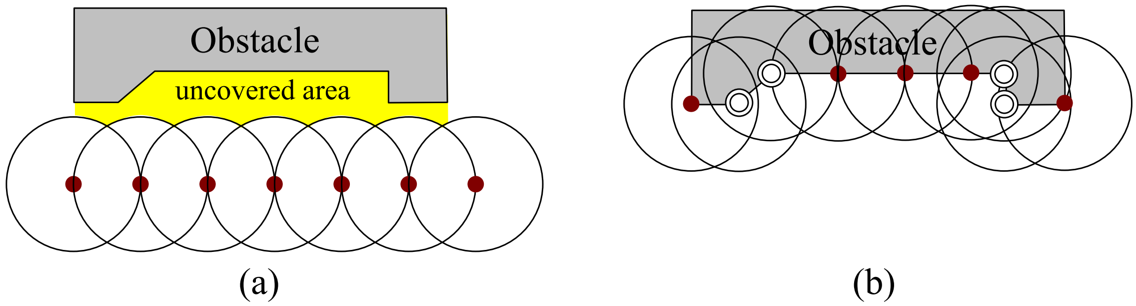


Figure 3.4: (a) uncovered area around an obstacle, and (b) extra sensors along the boundary to cover the uncovered area.

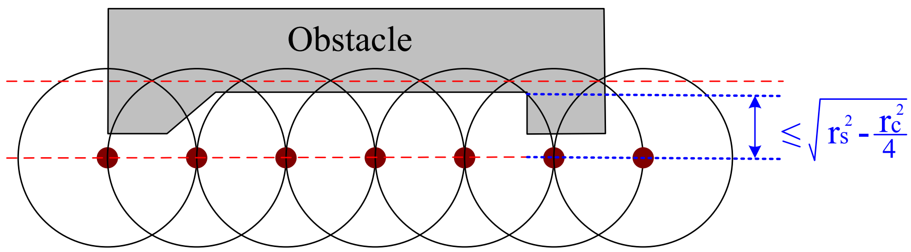


Figure 3.5: The case that the row of deployed sensors can fully cover the space near the boundary.

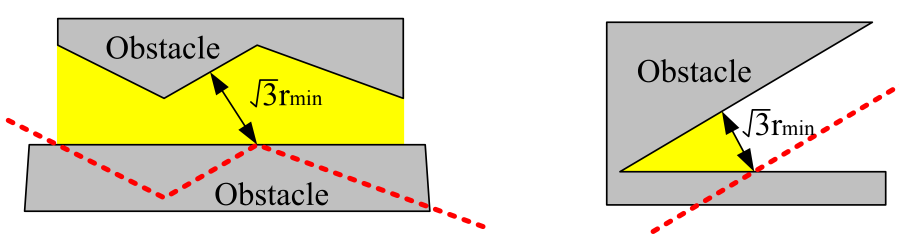
Note that some rows of deployed sensors may be very close to the boundaries so that they can fully cover the space near the boundaries. In such case, placing sensors along the boundaries may cause the waste of sensors since their coverage are completely overlapped by old ones. Therefore, if the distance between the row of deployed sensors and the boundary is no longer than $\sqrt{r_s^2 - \frac{r_c^2}{4}}$, we do not add extra sensors along such boundary. (Shown in Fig. 3.5.)

Chapter 4

Partitioning a Sensing Field

The results in Section 3 depend on partitioning 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 and A's boundaries by a distance of $\sqrt{3}r_{min}$. Such an expansion may cause overlapping with the original obstacles and A's boundary. For those parts with overlapping, we can take a projection back to the original perimeters to obtain some small regions. Taking Fig. 3.1(a) as an example, the dotted lines are expansion of A's boundaries. For those parts with overlapping, we can take a projection to obtain the small regions numbered 1 to 6 in Fig. 3.1(b). Fig. 4.1 shows two examples about the expansions of obstacles. Note that the above expansions may result in two different small regions. When such a conflict occurs, we can select the one that is larger as a small region.



(a) (b)

Figure 4.1: Two examples to find small regions. The dotted lines are expansions of obstacles.

Chapter 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. 5.1. We consider four cases: $(r_s, r_c) = (7, 5)$, $(5, 5)$, $(3.5, 5)$, and $(2, 5)$ to reflect the relationships of $r_s > r_c$, $r_s = r_c$, $r_s < r_c \leq \sqrt{3}r_s$, 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 number of sensors being used.

Fig. 5.2 compares the number of sensors being used when $r_c \leq \sqrt{3}r_s$ in different sensing fields. The connectivity-first method is dominated by the value of r_c, so the number of sensors is fixed when $r_c \leq \sqrt{3}r_s$. Thus, when $r_s \geq r_c$, this method uses the most sensors because the overlapping in coverage is very large. On the contrary, when $r_s < r_c \leq \sqrt{3}r_s$, 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. 5.3 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. 5.3.

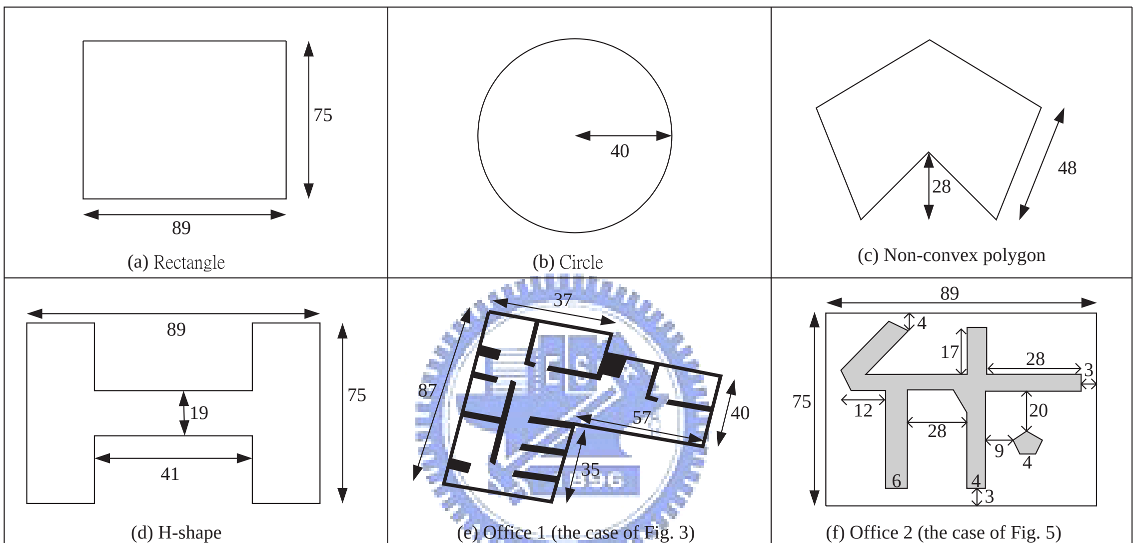
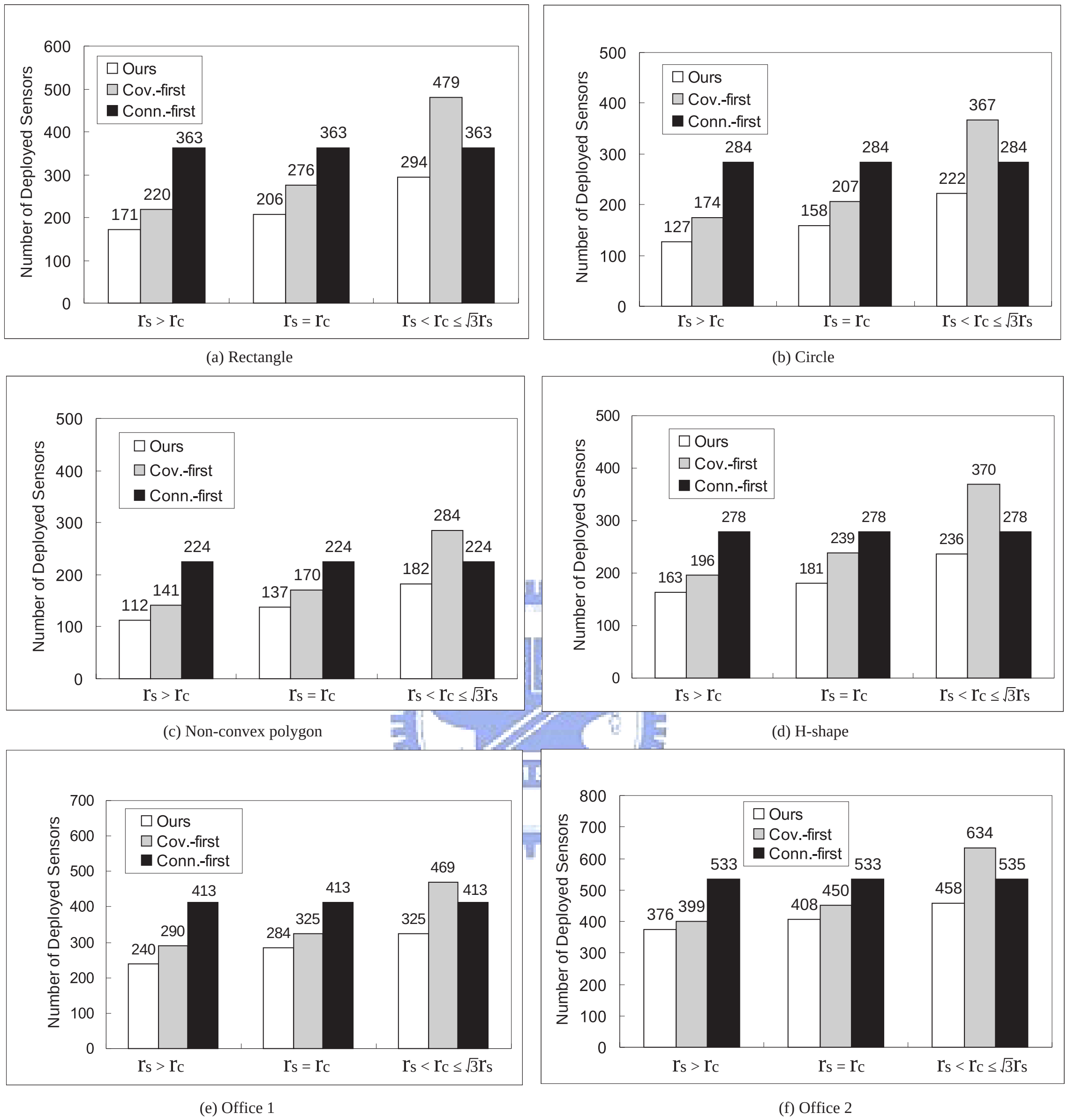


Figure 5.1: Sensing fields used in the simulations.



(a) Rectangle

(b) Circle

(c) Non-convex polygon

(d) H-shape

(e) Office 1

(f) Office 2

Figure 5.2: Comparison on the number of sensors used when $r_c \leq \sqrt{3}r_s$ under different shapes of sensing fields.

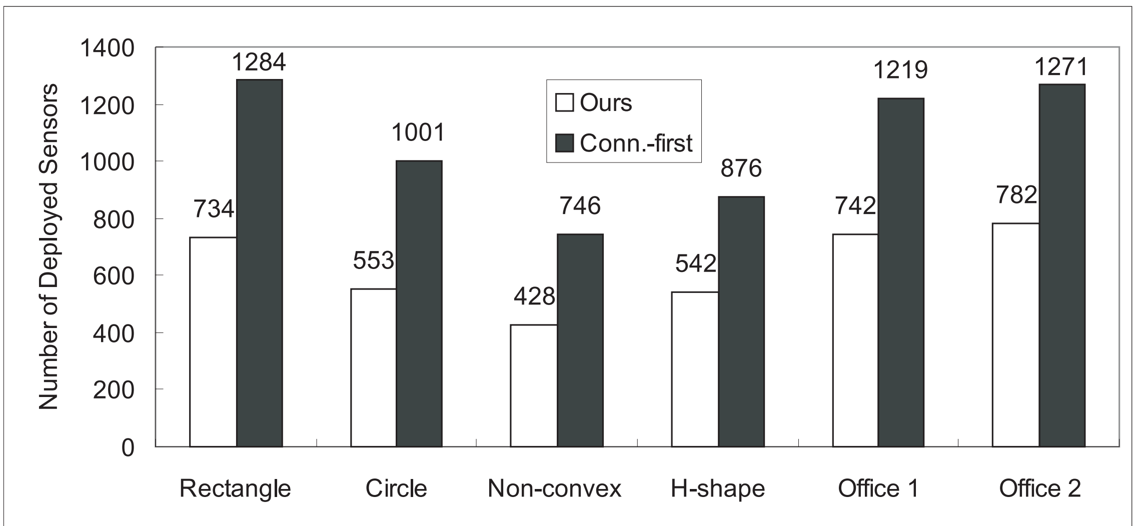


Figure 5.3: Comparison on the number of sensors used when $r_c > \sqrt{3}r_s$ under different shapes of sensing fields.

Chapter 6

Conclusions

In this work, we have proposed a systematical solution for sensor deployment. The sensing field is modeled as an arbitrary polygon with possible obstacles. Thus, the result may 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 distance r_c and sensing distance 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 3.2.1, we can add more columns of sensors among adjacent rows to improve the reliability of the network.

Bibliography

[1] A. Howard and M. J. Matarić and G. S. Sukhatme. Mobile sensor networks deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems*, 2002.

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

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

[4] P. Basu and J. Redi. Movement control algorithms for realization of fault-tolerant ad hoc robot networks. *IEEE Network*, 18:36–44, 2004.

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

[6] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy for target detection. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 42–48, 2002.

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

[8] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270, 1999.

[9] N. Heo and P. K. Varshney. Energy-efficient deployment of intelligent mobile sensor networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35:78–92, 2005.

[10] F. Hoffmann, M. Kaufmann, and K. Kriegel. The art gallery theorem for polygons with holes. In *Proceedings of 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 39–48, 1991.

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

[12] 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 (WSNA)*, pages 115–121, 2003.

[13] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for "Smart Dust". In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278, 1999.

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

[15] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, 2002.

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

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

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

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

[20] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 32–41, 2002.

[21] 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.

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

[23] F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy efficient robust sensing coverage in large sensor networks. *Computer Science Deparament - UCLA, Technical Report*, 2002.

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

[25] S. Zhou, M. Y. Wu, and W. Shu. Finding optimal placements for mobile sensors: wireless sensor network topology adjustment. In *Proceedings of the IEEE 6th Circuits and Systems Symposium*, volume 2, pages 529–532, 2004.

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

Curriculum Vita

Chun-Chi Hu (cchu@csie.nctu.edu.tw) received her B.S. degree in Computer Science from the National Chiao-Tung University, Taiwan, in 2003. Her research interests include wireless network and sensor network.