Efficient Deployment of Spinning, Directional Sensors for Temporal Coverage of Objects

You-Chiun Wang, Yung-Fu Chen, and Yu-Chee Tseng
Department of Computer Science
National Chiao-Tung University, Hsin-Chu, 30010, Taiwan
Email: {wangyc, yungfu, yctseng}@cs.nctu.edu.tw

Abstract

Because of application requirements or hardware designs, sensor nodes may possess sector-like coverage. On the other hand, with some machinery such as stepper motors, these sensor nodes can spin to cover the objects around them. We call such sensor nodes the spinning, directional (SD) sensors. *In the paper, we consider the* SD sensor deployment problem, *which decides how to deploy the minimum number of SD sensors to cover a set of static objects, such that each object is α-time covered, where $0 < \alpha < 1$. An object is α-*time covered *if it can be covered by at least one sensor for more than or equal to αT time within a fixed period (longer than T). The SD sensor deployment problem is NP-hard and therefore we propose a heuristic that adopts a greedy strategy, where we deploy each SD sensor that covers the maximum number of objects until all objects can be α-time covered. Simulation results verify that our deployment heuristic can significantly reduce the number of SD sensors to be deployed. The contributions of this paper are to define a new temporal coverage model to monitor objects by SD sensors and develop an efficient deployment heuristic.*

Keywords: directional sensor, spinning sensor, temporal coverage, wireless sensor network.

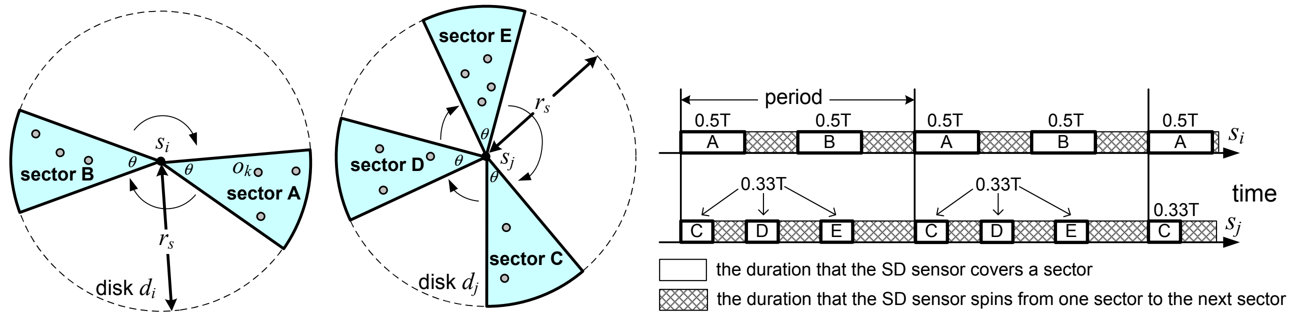
1. Introduction

With the capabilities of ad hoc communication and distributed processing, *wireless sensor networks (WSNs)* open a new frontier for pervasive environmental monitoring and therefore are widely adopted in many military and civil applications [1, 2, 3, 4, 5]. To guarantee that a WSN can well operate, sensor nodes have to be deployed to organize a connected network that covers either the whole sensing field or all the specific point-locations. In the literature, a large number of WSN deployment schemes [6, 7, 8] target at omnidirectional sensors with a disk-like sensing range. Nevertheless, in some practical WSN applications, sensor nodes may possess a sector-like sensing range because of application requirements or hardware designs. Furthermore, with some machinery such as stepper motors, these sensor nodes can possess certain mobility capabilities such as spinning [9, 10, 11]. We call such sensor nodes the *spinning, directional (SD) sensors*.

In this paper, we consider how to efficiently deploy SD sensors to monitor a set of static objects that are modeled by point-locations. We assume that all SD sensors can be precisely deployed at any location within the sensing field. The coverage area of each SD sensor is modeled by a sector and the SD sensor can spin to scan a whole disk, as illustrated in Fig. 1(a). The time axis is divided into fixed periods. During each period, an SD sensor will spin one cycle and stop to detect all objects for a total (constant) time T, as illustrated in Fig. 1(b). An object is said to be *α-time covered* if during each period, this object is within the sensing range of at least one SD sensor for more than or equal to αT time, where $0 < \alpha < 1$. A network is said to achieve *α-time coverage* if all monitored objects can be α-time covered. Fig. 1(a) and (b) together illustrate an example, where the objects within disks d_i and d_j are 0.5-time and 0.33-time covered, respectively. In this case, the network is said to achieve 0.33-time coverage. The above sensing model of SD sensors can be used in many WSN applications. One typical example is the visual surveillance system using spinning video cameras. According to the above sensing model, we target at the *SD sensor deployment problem*, which decides how to deploy the minimum number of SD sensors to cover a set of objects such that the network achieves α-time coverage.

The SD sensor deployment problem is NP-hard because the *geometric disk cover (GDC)* problem [12], which is a well-known NP-hard problem, is its one special instance. In particular, given a set of point-locations, the GDC problem decides how to place the minimum number of disks to cover these point-locations. Consider that each SD sensor can cover a sector with angle of $\theta \in (0, \pi)$. By setting $\alpha = \frac{\theta}{2\pi}$, the SD sensor deployment problem becomes the same as the GDC problem because each SD sensor now can cover a whole disk in every period.

In this paper, we propose an efficient heuristic to solve the SD sensor deployment problem. The idea of our heuristic is to first place the minimum number of disks to cover all objects. Then, we always place one SD sensor on the



(a)

(b)

Figure 1. (a) The sensing model of SD sensors and (b) the temporal coverage model of objects.

center of the disk that can cover the maximum number of objects, until all objects are α-time covered. Finally, we deploy some additional sensor nodes to maintain the network connectivity. The contributions of this paper are three-folds:

1. We define a new SD sensor deployment problem that allows directional sensors to spin to guarantee temporal coverage of objects, which could be used in many WSN applications such as surveillance.

2. Because the SD sensor deployment problem is NP-hard, we propose an efficient deployment heuristic by adopting the greedy strategy to reduce the number of SD sensors to be deployed. In this way, we can significantly reduce the network deployment cost. Simulation results also verify the effectiveness of the proposed heuristic.

3. Our deployment heuristics allow an arbitrary relationship between the sensing distance and the communication distance of SD sensors. In this way, it can satisfy different application requirements.

We organize the rest of this paper as follows. Section 2 surveys some related work on the research field of sensor deployment and directional sensor networks. Section 3 formally defines the SD sensor deployment problem. Our efficient deployment heuristic is proposed in Section 4. In Section 5, we present the simulation results. Conclusions and future work are drawn in Section 6.

2. Related Work

In the literature, the issue of how to deploy omnidirectional sensor nodes to organize a WSN has been extensively investigated [13]. Many studies discuss how to use mobile sensors to automatically organize a network. For instance, the research efforts in [14, 15, 16] move sensor nodes to improve the network coverage by adopting a Voronoi diagram or the attractive/repulsive forces among sensor nodes. The work in [17] divides the sensing field into multiple grids, and then moves sensor nodes from those grids with more sensor nodes to those grids with fewer sensor nodes to result in a more uniform network topology. The studies in [18, 19] address two deployment-related problems: the *sensor placement problem* and the *sensor dispatch problem*. The sensor placement problem decides how to place the minimum number of sensor nodes to cover the entire sensing field while the sensor dispatch problem decides how to move sensor nodes to the target locations calculated by the placement solution such that their moving energy can be reduced.

Several research efforts consider directional sensor networks where sensor nodes are arbitrarily deployed. Given a set of point-locations, the work in [20] addresses how to identify a minimum set of directions to cover the maximum number of point-locations. The study in [21] analyzes the probability that a point-location can be covered by directional sensors. The work in [22] divides the network into multiple subsets of directional sensors to alternatively cover a given set of point-locations so that the system lifetime can be extended.

The issue of deploying fixed directional sensors is also discussed in the literature. For instance, the work in [23] addresses deploying a minimum number of directional sensors to organize a connected network that covers either the whole sensing field or a giving set of point-locations. Given a set of point-locations, the study in [24] attempts to minimize the number of deployed directional sensors by adopting an integer linear programming solution. In the work of [25], the issue of using SD sensors to localize objects is discussed, but the spinning range of each SD sensor is constrained by a limited angle. As can be seen, none of existing work discusses the SD sensor deployment problem and the temporal coverage model addressed in this paper. We will develop an efficient deployment heuristic to solve the SD sensor deployment problem.

3. The SD Sensor Deployment Problem

Consider that there is a set of static objects $\mathcal{O} = \{o_1, o_2, \cdots, o_m\}$ to be monitored by SD sensors, where each object is modeled by a point-location in a 2D plane. Each SD sensor s_i possesses a sensing range modeled by a *sector* with angle of $\theta \in (0, \pi)$ and radius of r_s, and an omnidirectional communication range with radius of r_c, where

r_s and r_c are the sensing distance and communication distance of sensor nodes, respectively. We make no assumption on the relationship between r_s and r_c. An object o_k is said to be *covered* by an SD sensor s_i if object o_k locates within s_i's sensing range, as illustrated in Fig. 1(a). All SD sensors have the spinning capability. When an SD sensor s_i spins one cycle (without changing its position), its sensing range will scan a whole disk d_i that is centered at s_i and with radius of r_s. According to the objects within disk d_i, we can divide d_i into p disjointed sectors, where each sector has an angle of θ. SD sensor s_i then spins its sensing range to fit each of these p sectors. When the sensing range of SD sensor s_i completely fits a sector, we say that s_i *covers* that sector. Fig. 1(a) illustrates two examples. Disk d_i is divided into two sectors and SD sensor s_i will spin to cover sectors A and B, while disk d_j is divided into three sectors and SD sensor s_j will spin to cover sectors C, D, and E.

The time axis is divided into fixed-length *periods*. During each period, an SD sensor s_i will spin one cycle to cover all the objects within its corresponding disk d_i. Specifically, for each disk d_i with p sectors, its corresponding SD sensor s_i will stay to cover each sector for $\frac{T}{p}$ time and then spin to cover the next sector. Therefore, the length of each period is the total time that an SD sensor stays to cover all the sectors (that is, time T) and the time that the SD sensor needs to spin one cycle (without stopping). Note that strict time synchronization among SD sensors is not necessary. Fig. 1(b) illustrates two examples. Since disks d_i and d_j are divided into two and three sectors, SD sensors s_i and s_j will stay to cover each of their sectors for $0.5T$ and $0.33T$ time, respectively.

Given a threshold α, where $0 < \alpha < 1$, an object is said to be *α-time covered* if and only if during each period, this object can be covered by at least one SD sensor for more than or equal to αT time. Fig. 1(b) illustrates two example, where the objects within disks d_i and d_j are 0.5-time and 0.33-time covered, respectively. Then, given the set of objects $\mathcal{O}$ and the threshold α, the *SD sensor deployment problem* decides how to deploy the minimum number of SD sensors to cover all objects in $\mathcal{O}$ such that each object is α-time covered. Note that in this case, each SD sensor can cover at most $\lfloor \frac{1}{\alpha} \rfloor$ sectors during each period. When a disk is divided into more than $\lfloor \frac{1}{\alpha} \rfloor$ sectors, it requires more than one SD sensor to cover all of its sectors. Fig. 1(a) illustrates an example. Assuming that each object in disk d_j should be 0.5-time covered (that is, $\alpha = 0.5$), we have to place two SD sensors at disk d_j to guarantee that all of its objects can satisfy 0.5-time covered.

4. The Proposed Deployment Heuristic

Given a set of objects $\mathcal{O}$ to be covered by SD sensors, our deployment heuristic consists of the following four phases:

- **Phase 1:** We find a set of disks $\mathcal{D}$ that can cover all objects in $\mathcal{O}$.

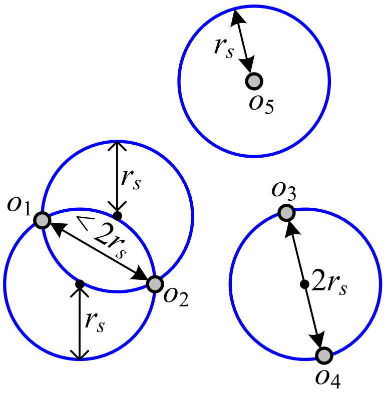


Figure 2. Examples of finding a set of disks $\mathcal{D}$ to cover all objects by the modified GDC scheme.

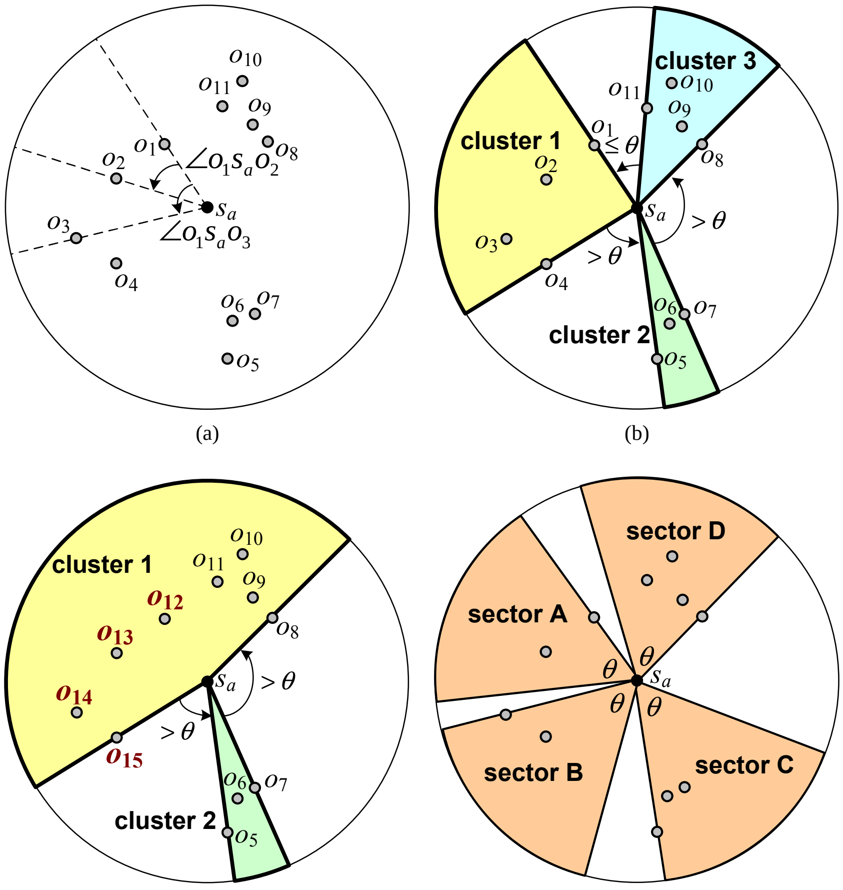
- **Phase 2:** For each disk in $\mathcal{D}$, we calculate the minimum number of sectors that cover all objects in that disk.
- **Phase 3:** We then place SD sensors at a subset of disks in $\mathcal{D}$ to cover all objects in $\mathcal{O}$ such that each object can be α-time covered.
- **Phase 4:** We finally deploy the minimum number of additional sensor nodes to maintain the network connectivity.

Below, we detail the operations of each phase. In phase 1, we modify the GDC scheme in [12] to calculate the set of disks $\mathcal{D}$ that cover all objects in $\mathcal{O}$. This modified GDC scheme involves the following three steps:

1. For any two objects, say, o_i and o_j in $\mathcal{O}$, if their Euclidean distance is smaller than $2r_s$, we place two disks such that their circumferences intersect at both objects o_i and o_j.
2. For any two objects, say, o_i and o_j in $\mathcal{O}$, if their Euclidean distance is equal to $2r_s$, we place one disk such that its circumference passes through both objects o_i and o_j.
3. For any object, say, o_k in $\mathcal{O}$ whose Euclidean distance to its nearest object is larger than $2r_s$, we place one disk such that the disk's center locates at object o_k.

Fig. 2 illustrates three examples of the above steps. Using the above modified GDC scheme, we can find a set of disks $\mathcal{D}$ whose size is no larger than $2C_2^m$.

In phase 2, we need to calculate the minimum number of sectors in each disk to cover all the objects within that disk. Here, we propose a *sector dividing scheme* to find the sectors in each disk. The idea is to first identify where objects gather in the disk and group them into clusters accordingly. Then, we place sectors to cover all objects in each cluster. In particular, supposing that there are K objects within the



(a) (b) (c) (d)

Figure 3. The sector dividing scheme on a disk: (a) assign indices to objects, (b) group objects into clusters, (c) merge the first cluster and the last cluster if their included angle is no larger than θ, and (d) place sectors to cover the objects within the disk.

disk, the sector dividing scheme on that disk is composed of the following three steps:

1. We arbitrarily select one object, say, o_1 within the disk as the initial object. Then, we scan all the objects in the disk in a counterclockwise direction and assign indexes to them accordingly. Specifically, suppose that s_a is the disk's center. Object o_i is assigned with a smaller index than object o_j if and only if $\angle o_1 s_a o_i < \angle o_1 s_a o_j$. Note that all angles are scanned in a counterclockwise direction. However, when there is a tie, we arbitrarily assign indexes to these objects. Fig. 3(a) illustrates an example. Because $\angle o_1 s_a o_2 < \angle o_1 s_a o_3$, object o_2 is assigned with a smaller index than object o_3.

2. Beginning from object o_1, we scan all objects according to their indexes. We then group all objects into clusters. In particular, object o_1 is added into cluster 1. For two adjacent objects o_i and o_{i+1}, where $1 \leq i < K$, supposing that o_i is added into cluster j, then object o_{i+1} is also added into cluster j if $\angle o_i s_a o_{i+1} \leq \theta$. Otherwise, o_{i+1} is added into a new cluster $j+1$. Fig. 3(b) illustrates an example, where there are three clusters calculated. After grouping all objects, we then check whether or not the first cluster and the last cluster can be merged. Specifically, if $\angle o_K s_a o_1 \leq \theta$, these two clusters can be merged and we assign new indexes to those objects originally in cluster 1. In particular, supposing that objects o_1, o_2, $\cdots$, and o_l are originally added into cluster 1, we assign new indexes o_{K+1}, o_{K+2}, $\cdots$, and o_{K+l} to them, respectively. Fig. 3(c) illustrates an example, where we merge clusters 1 and 3 into the new cluster 1, and give the new indexes o_{12}, o_{13}, o_{14}, and o_{15} to those objects originally indexed with o_1, o_2, o_3, and o_4, respectively.

3. Finally, we place sector(s) to cover each cluster of objects. In particular, beginning from the object with the smallest index in the cluster, say, o_{small}, we place one sector whose edge passes through o_{small} such that this sector can cover the maximum number of objects in the corresponding cluster. Then, we remove all the objects covered by that sector from the cluster. The above two operations are repeated until all the objects in the

cluster are removed. Fig. 3(d) illustrates an example, where there are four sectors A, B, C, and D calculated to cover the objects in clusters 1 and 2.

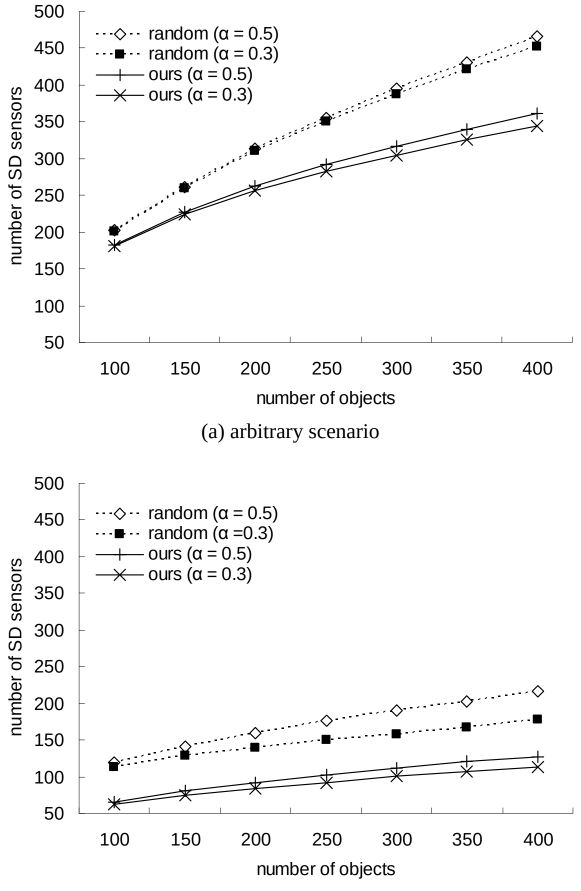
We remark that after conducting the sector dividing scheme on a disk, say, d_i, all objects within disk d_i will be removed from $\mathcal{O}$. Therefore, even though two disks overlap with each other, their covered objects are considered as *disjointed*. We can sort disks in $\mathcal{D}$ according to their objects in a decreasing order and then execute the sector dividing scheme on these disks following the sorting sequence. In this way, we can execute the sector dividing scheme on at most $O(m)$ disks (because there are m objects in $\mathcal{O}$). In this way, all remaining disks (without executing the sector dividing scheme) can be removed from $\mathcal{D}$ and therefore the size of $\mathcal{D}$ is shrunken to $O(m)$.

In phase 3, we deploy SD sensors to cover objects. Specifically, we select the disk, say, d_i that covers the maximum number objects when we deploy an SD sensor on it. Here, when a disk contains more than $\lfloor \frac{1}{\alpha} \rfloor$ sectors, the maximum number of objects covered by an SD sensor is the number of objects in the first $\lfloor \frac{1}{\alpha} \rfloor$ sectors, where sectors are sorted according to their numbers of covered objects in a decreasing order. Then, we deploy an SD sensor at d_i's center and make the SD sensor spin to cover each of d_i's p sectors. Note that when disk d_i has more than $\lfloor \frac{1}{\alpha} \rfloor$ sectors, these p sectors will be the first $\lfloor \frac{1}{\alpha} \rfloor$ sectors that cover the maximum number of objects. Otherwise, p will be the total number of sectors in disk d_i. Then, we remove the objects covered by the SD sensor from the set $\mathcal{O}$. In this way, these p sectors are also removed from disk d_i. The above operations are repeated until the set $\mathcal{O}$ becomes empty. Fig. 3(d) illustrates an example, where we set $\alpha = 0.5$. In this case, we first deploy an SD sensor at location s_a to cover sectors C and D and remove these two sectors from the disk. Then, we place another SD sensor at location s_a again to cover sectors A and B. Note that each SD sensor will stay to cover each of its corresponding sector for $0.5T$ time.

Finally, in phase 4, we will deploy some additional sensor nodes to maintain the network connectivity. Several research efforts [23, 26] have also discussed this issue. Here, we modify the scheme in [23]. Specifically, given a set of sensor nodes $\mathcal{S}$ calculated by our deployment heuristic, we first calculate the minimum spanning tree whose tree nodes are all the sensor nodes in $\mathcal{S}$. Then, for each tree edge whose length, say, L is larger than the communication distance r_c, we deploy $\left(\frac{L}{r_c} - 1\right)$ additional sensor nodes along that tree edge. The distance between two adjacent additional sensor nodes is r_c. In this way, the network connectivity along that tree edge can be maintained. We repeat the above operations until all tree edges are examined. In this way, we can maintain the entire network connectivity.

5. Performance Evaluations

To verify the effectiveness of our deployment heuristic, we develop a simulator in C++. The sensing field is modeled by a rectangle with area of 400×400, on which there are some static objects needed to be monitored. Two scenarios are considered. In the *arbitrary* scenario, objects are arbitrarily placed inside the sensing field. In the *regular* scenario, we select ten point-locations inside the sensing field and then place objects around these point-locations. For comparison, we also develop a *random deployment scheme*. In this scheme, we arbitrarily select a subset of objects as the centers of disks such that these disks can cover all objects. Then, we execute the sector dividing scheme on these disks and place SD sensors accordingly. Each SD sensor possesses a sensing distance r_s of 10 and a communication distance r_c of 20.

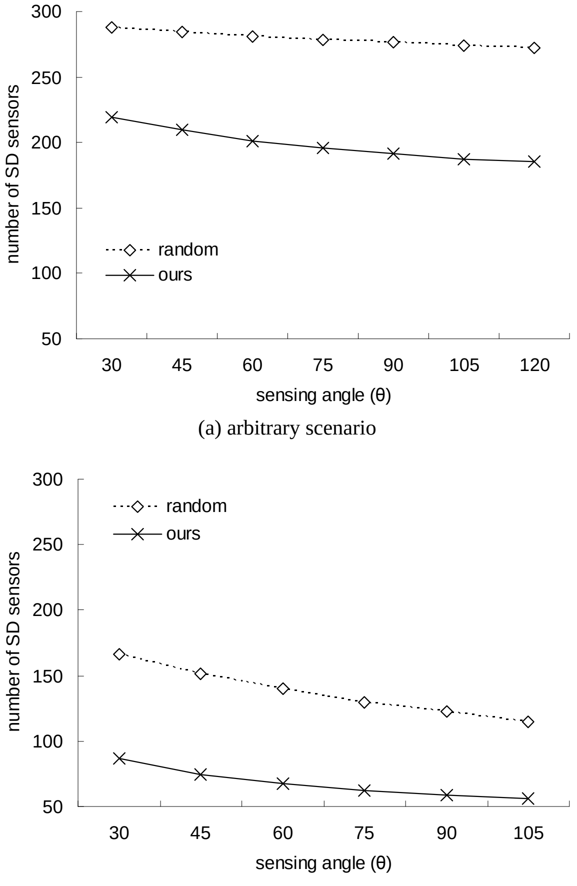


(a) arbitrary scenario

(b) regular scenario

Figure 4. The number of SD sensors needed to be deployed under different numbers of objects.

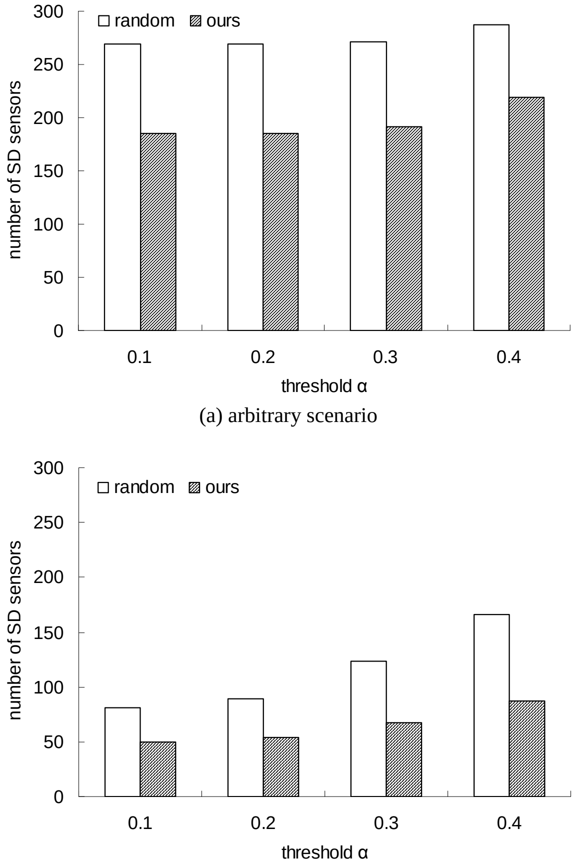
Fig. 4 illustrates the number of SD sensors used to monitor objects by the random deployment scheme and our deployment heuristic. Here, the number of deployed SD sensors includes the number of SD sensors used to cover objects and the number of additional sensor nodes used to maintain the network connectivity. The α value is set to 0.5 and 0.3 so that each SD sensor can cover at most two and three sectors, respectively. The sensing angle θ is set to 45°. The number of objects is set to 100, 150, 200, 250,



(a) arbitrary scenario

(b) regular arbitrary scenario

Figure 5. The number of SD sensors needed to be deployed under different values of the sensing angle θ.



(a) arbitrary scenario

(b) regular scenario

Figure 6. The number of SD sensors needed to be deployed under different values of the threshold α.

300, 350, and 400. From Fig. 4, we observe that the number of deployed SD sensors increases when the number of objects increases, because we need more SD sensors to cover objects. A smaller α value can help reduce the number of deployed SD sensors because each SD sensor can now cover more sectors in every period. We can observe that under both scenarios, our deployment heuristic can significantly reduce the number of deployed SD sensors, as compared with the random deployment scheme.

We then evaluate the effect of different values of the sensing angle θ on the number of deployed SD sensors, as illustrated in Fig. 5. Because changing the sensing angle would not affect the number of additional sensor nodes used to maintain the network connectivity, we measure only the number of SD sensors used to cover objects (so the number of sensor nodes will be fewer than those in Fig. 4). The number of objects is set to 400 and we set the threshold $\alpha = 0.5$. The sensing angle θ is set to 30°, 45°, 60°, 75°, 90°, and 120°. Explicitly, a larger sensing angle θ indicates that each SD sensor can cover a wider range and therefore more objects may be covered by the SD sensor. In this case, the number of deployed SD sensors decreases when the sensing angle θ increases. From Fig. 5, we observe that our deployment heuristic can deploy the minimum number of SD sensors under the arbitrary and regular scenarios.

Finally, we evaluate the effect of different values of the threshold α on the number of deployed SD sensors, as illustrated in Fig. 6. Similarly, we consider only the number of SD sensors used to cover objects because changing the value of α threshold does not affect the number of additional sensor nodes used to maintain the network connectivity. The number of objects is set to 400 and the sensing angle θ is set to 30°. The threshold α is set to 0.1, 0.2, 0.3, and 0.4 so that each SD sensor can cover at most 10, 5, 3, and 2 sectors in every period. Note that when $\alpha = 0.5$, each SD sensor can cover at most two sectors in every period (same as the case of $\alpha = 0.4$) and when $\alpha > 0.5$, each SD sensor can cover only one sector in every period. Therefore, we do not observe the simulation results when $\alpha \geq 0.5$. Recall that each SD sensor can cover at most $\lfloor \frac{1}{\alpha} \rfloor$ sectors. Therefore, a larger α value will increase the number of deployed SD sensors. From Fig. 6, it can be observed that our deployment heuristic can deploy fewer SD sensors under different threshold α, as compared with the random deployment scheme.

6. Conclusions and Future Work

In this paper, we define a new SD sensor deployment problem to achieve temporal coverage of objects and propose an efficient heuristic to solve the deployment problem. Our deployment heuristic deploys SD sensors at those disks that cover the maximum number of objects. A solution to maintain the network connectivity by adopting a minimum spanning tree is also proposed. Simulation results show that the proposed deployment heuristic can reduce the number of SD sensors under different situations. For future work, we will investigate how to make SD sensors cooperate with each other to monitor objects such that the number of deployed SD sensors can be further reduced. In addition, we plan to develop some surveillance applications by SD sensors.

Acknowledgement

Y.-C. Tseng's research is co-sponsored by MoE ATU Plan, by NSC grants 97-3114-E-009-001, 97-2221-E-009-142-MY3, 98-2219-E-009-019, and 98-2219-E-009-005, by MOEA 98-EC-17-A-02-S2-0048, and 98-EC-17-A-19-S2-0052, by ITRI, Taiwan, by III, Taiwan, and by Intel.

References

[1] C. Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, no. 8, pp. 1247–1256, 2003.

[2] Y. C. Tseng, Y. C. Wang, K. Cheng, and Y. Y. Hsieh, "iMouse: an integrated mobile surveillance and wireless sensor system," *IEEE Computer*, vol. 40, no. 6, pp. 60–66, 2007.

[3] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung, and J. H. Park, "Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 400–411, 2009.

[4] L. W. Yeh, Y. C. Wang, and Y. C. Tseng, "iPower: an energy conservation system for intelligent buildings by wireless sensor networks," *International Journal of Sensor Networks*, vol. 5, no. 1, pp. 1–10, 2009.

[5] S. C. Hu, Y. C. Wang, C. Y. Huang, and Y. C. Tseng, "A vehicular wireless sensor network for CO_2 monitoring," *Proc. IEEE Conference on Sensors*, pp. 1498–1501, 2009.

[6] F. Y. S. Lin and P. L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks," *IEEE Communication Letters*, vol. 9, no. 1, pp. 43–45, 2005.

[7] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," *Proc. IEEE Wireless Internet Conference*, pp. 114–121, 2005.

[8] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 131–142, 2006.

[9] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only SLAM for robots operating cooperatively with sensor networks," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2078–2084, 2006.

[10] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Network*, vol. 51, no. 4, pp. 921–960, 2007.

[11] S. Aoki, J. Nakazawa, and H. Tokuda, "Spinning sensors: a middleware for robotic sensor nodes with spatiotemporal models," *Proc. IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 89–98, 2008.

[12] B. Xiao, Q. Zhuge, Y. He, Z. Shao, and E. H. M. Sha, "Algorithms for disk covering problems with the most points," *Proc. IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 541–546, 2003.

[13] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

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

[15] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *Proc. IEEE INFOCOM*, pp. 2469–2479, 2004.

[16] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, 2005.

[17] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," *Proc. IEEE INFOCOM*, pp. 2302–2312, 2005.

[18] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[19] Y. C. Wang and Y. C. Tseng, “Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[20] J. Ai and A. A. Abouzeid, “Coverage by directional sensors in randomly deployed wireless sensor networks,” *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, 2006.

[21] H. Ma and Y. Liu, “Some problems of directional sensor networks,” *International Journal of Sensor Networks*, vol. 2, no. 1/2, pp. 44–52, 2007.

[22] Y. Cai, W. Lou, M. Li, and X. Y. Li, “Energy efficient target-oriented scheduling in directional sensor networks,” *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1259–1274, 2009.

[23] X. Han, X. Cao, E. L. Lloyd, and C. C. Shen, “Deploying directional sensor networks with guaranteed connectivity and coverage,” *Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 153–160, 2008.

[24] Y. Osais, M. St-Hilaire, and F. R. Yu, “On sensor placement for directional wireless sensor networks,” *Proc. IEEE International Conference on Communications*, 2009.

[25] J. Y. Lee, “Exploiting constrained rotation for localization of directional sensor networks,” *Proc. IEEE International Wireless Communications and Mobile Computing Conference*, pp. 767–772, 2008.

[26] S. Zhou, M. Y. Wu, and W. Shu, “Finding optimal placements for mobile sensors: wireless sensor network topology adjustment,” *Proc. IEEE Circuits and Systems Symposium Emerging Technologies: Frontiers of Mobile and Wireless Communications*, pp. 529–532, 2004.