k-Angle Object Coverage Problem in a Wireless Sensor Network

Yu-Chee Tseng, *Fellow, IEEE*, Po-Yu Chen, *Member, IEEE*, and Wen-Tsuen Chen, *Fellow, IEEE*

***Abstract*— One of the fundamental issues in sensor networks is the *coverage problem*, which reflects how well a field is monitored or tracked by sensors. Various versions of this problem have been studied, such as object, area, barrier, and hole coverage problems. In this paper, we define a new *k-angle object coverage problem* in a wireless sensor network. Each sensor can only cover a limited angle and range, but can freely rotate to any direction to cover a particular angle. Given a set of sensors and a set of objects at known locations, the goal is to use the least number of sensors to *k-angle-cover* the largest number of objects such that each object is monitored by at least k sensors satisfying some angle constraint. We propose centralized and distributed polynomial-time algorithms to solve this problem. Simulation results show that our algorithms can be effective in maximizing coverage of objects. A prototype system is developed to demonstrate the usefulness of angle coverage.**

***Index Terms*— Coverage problem, pervasive computing, sensor network, video surveillance, wireless network.**

I. Introduction

A WIRELESS SENSOR NETWORK (WSN) consists of many inexpensive wireless nodes, each capable of collecting, storing, and processing environmental information, and communicating with neighboring nodes. Recently, MAC protocols [1], [2], routing and transport protocols [3], [4], and localization technologies [5]–[7] have been studied for WSNs.

One fundamental issue in WSNs is the *coverage problem*, which reflects how well a field or targets are monitored or tracked by sensors. Two related problems have been studied in computational geometry. The *art gallery problem* [8] tries to determine the minimum number of rotatable cameras to monitor an environment. The *circle covering problem* [9] intends to use the minimum number of unit disks to fully cover a rectangle. The *area coverage problem* in WSNs is dealt by [10] as a decision problem. The *object coverage problem* is studied in [11]. References [12]–[16] deal with both coverage and connectivity simultaneously. Energy-conserving coverage issues are addressed in [17]–[22]. Recently, solutions for *barrier coverage* [23], [24] and *hole coverage* [25] are proposed.

Only few works [26]–[29] investigate the coverage problem with directional sensors that only can cover a sector area (as opposed to omnidirectional sensors). Reference [26] proposes to divide the original circular covering region into eight sectors, each to be covered by one directional sensor. A *Maximum Coverage with Minimum Sensors (MCMS)* problem is defined to maximize the number of objects that are 1-covered and an integer linear programming (ILP) formulation and a greedy heuristic are proposed. Several camera-coverage problems are addressed with some exponential-time algorithms are presented in [27]. In [28], a selecting and orienting d-sensors for k-coverage problem is modeled and a greedy algorithm to select and orient the minimal number of directional sensors to k-cover a set of targets is proposed. Reference [29] models the sensing field by a set of points, on which directional sensors can be deployed. Given a subset of critical points, the work shows how to deploy the minimum number of sensors to cover all critical points by integer linear programming. However, once sensors are deployed, they can not rotate anymore. All these works deal with directional sensors to extend the traditional coverage problems; however, the angle(s) from which an object is covered is still not well elaborated (refer to Fig. 1 and Fig. 2).

In this work, we define a new *k-angle object coverage problem*, where k is an integer. In contrast to previous studies, where sensors are assumed to be able to cover 360 degrees, we assume that each sensor can only monitor a specific angle within a limited range. Practically, one may imagine using video sensors for surveillance purposes. Such sensors are rotatable but can only cover a limited angle at a time. Further, to clearly monitor an object, we enforce that it must be simultaneously monitored by at least k sensors from multiple angles satisfying certain angle constraints (to be defined later). Several new applications may be triggered by this problem (see Section II). Given a set of sensors and a set of objects, our goal is to use the minimum number of sensors to k-angle cover the maximum number of objects. We propose two heuristics. The first scheme tries to fix the set of sensors that benefit the coverage levels of the most objects first. The second one evaluates the contribution of each sensor when facing each angle. Then the one which can help increase the coverage levels of the highest-covered objects is selected first. After all objects are k-angle-covered,

Manuscript received October 27, 2011; revised March 6, 2012; accepted April 17, 2012. Date of publication May 4, 2012; date of current version November 2, 2012. This work was supported in part by the National Science Council, Taiwan, under Grant 100-2219-E-009-001 and Grant 100-2219-E-007-001, the Industrial Technology Research Institute of Taiwan, and D-Link Inc. The associate editor coordinating the review of this paper and approving it for publication was Prof. Weileun Fang.

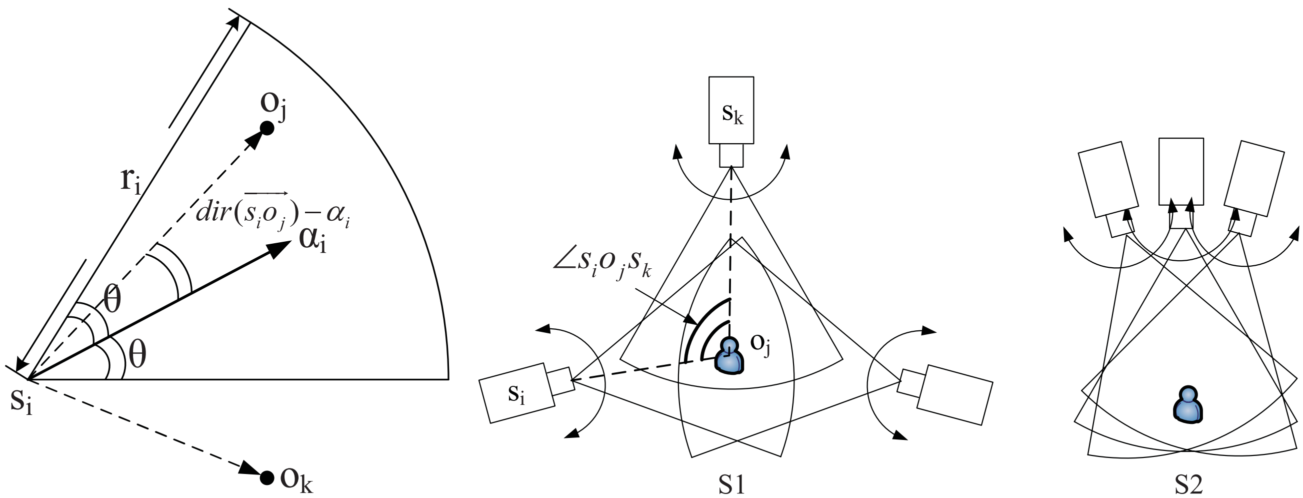
Y.-C. Tseng is with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: yctseng@cs.nctu.edu.tw).

P.-Y. Chen is with the Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan (e-mail: jaa@mnet.cs.nthu.edu.tw).

W.-T. Chen is with the Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, and also with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: wtchen@cs.nthu.edu.tw).

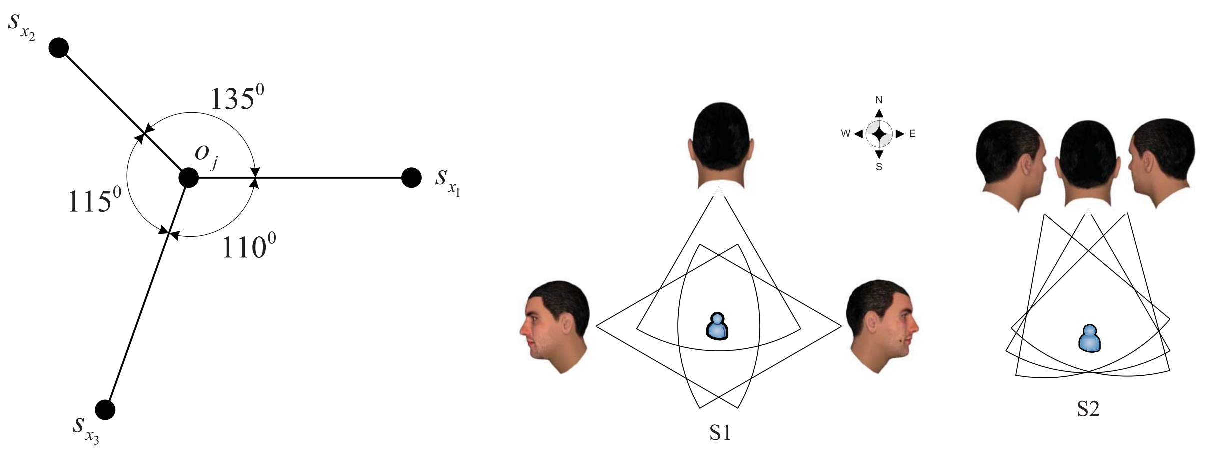
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2012.2198054



(a) (b)

Fig. 1. Example of angle coverage.



(a) (b)

Fig. 2. $(3, \pi/2)$-angle-covered example.

the redundant sensors can enter low-power mode to save energy.

Section II formally defines the k-angle object coverage problem. Our solutions and some extensions are presented in Section III and Section IV, respectively. Section V presents our simulation results. Section VI presents our implemental experiences. Finally, Section VII concludes this work.

II. Problem Statement

We are given a set of n sensors, $S = \{s_1, s_2, \ldots, s_n\}$, and a set of m objects, $O = \{o_1, o_2, \ldots, o_m\}$, in a two-dimensional area A. Sensors' and objects' locations are all known in advance. The location and sensing range of s_i, $i = 1, \ldots, n$, are denoted by (x_i, y_i) and r_i, respectively. Sensors, valid ranges are directional and each sensor can be rotated to any particular direction to cover an angle range of 2θ. Specially, each s_i can rotate to any direction $\alpha_i \in [0, 2\pi)$ and cover the angle between $\alpha_i - \theta$ and $\alpha_i + \theta$ within a distance of r_i. For example, in Fig. 1(a), object o_j is covered by s_i, but o_k is not.

When monitoring an object, we usually desire to observe it from multiple angles so as to clearly capture its behavior. Fig. 1(b) shows two ways to monitor an object from three angles. Scenario S1 is more favorable because we can extract more complete features of the object from different directions. In contrast, S2 provides observations that are likely duplicated.

Definition 1: Given any $s_i \in S$ and any $o_j \in O$, the distance between s_i and o_j is denoted by $dis(s_i, o_j)$, the vector from the location of s_i to the location o_j is denoted by $\overrightarrow{s_i o_j}$, and the direction of $\overrightarrow{s_i o_j}$ is denoted by $dir(\overrightarrow{s_i o_j})$. Given the current direction α_i of s_i, we say that o_j is *angle-covered* by s_i if $dis(s_i, o_j) \le r_i$ and $-\theta \le dir(\overrightarrow{s_i o_j}) - \alpha_i \le \theta$.

Definition 2: The angle formed by sensor s_i, object o_j, sensor s_k is denoted by $\angle s_i o_j s_k$. Given an integer k and an angle of separation ω, we say that an object o_j is (k, ω)*-angle-covered* if there is a sequence of k sensors $s_{x_1}, s_{x_2}, \ldots, s_{x_k}$ each angle-covering o_j and $\angle s_{x_p} o_j s_{x_{p+1}} \ge \omega$ for $p = 1 \ldots k$ (for simplicity we regard $s_{x_{k+1}}$ as s_{x_1}).

Fig. 2(a) shows an example, where o_j is $(3, \frac{\pi}{2})$-angle-covered by sensors s_{x_1}, s_{x_2}, and s_{x_3}. Since $\omega = 90°$, it is guaranteed that no angle larger than $360° - 2\omega = 180°$ of the object is not covered by any sensor. In Fig. 2(b), if the man is facing toward the south, S1 can still get clear views of his face. However, in this case, S2 can not get clear views of his face. It is hard to prove the following lemma, which indicates the largest angle that may not be covered under our constraint.

Lemma 1: If an object o_j is (k, ω)-*angle-covered*, then there is no angle larger than $2\pi - (k-1)\omega$ of the object that is not covered by any sensor.

Definition 3: Given k, θ, ω, S, and O, the (k, ω)-*Angle Object Coverage Problem* is to use the minimum number of sensors by tuning their directions to (k, ω)-angle-cover the maximum number of objects.

The (k, ω)-angle object coverage problem may have many applications. Examples include multi-camera video surveillance and motion capture [30]–[32]. Multi-lateration localization, such as those based on the angle-of-arrival model, is also related to this issue when there are multiple objects and sensors [7], [33]. Another possibility is vehicle-tracking applications.

Theorem 1: The (k, ω)-angle object coverage problem is NP-complete.

Proof: We prove its *NP*-completeness property by showing its special case of $\theta = 2\pi$ and $\omega = 0°$ to be *NP*-complete. For any given number of sensors, the problem of deriving the maximum number of objects to be covered with the minimum number of sensors can be treated as the Maximum Coverage Problem [34], which is known to be *NP*-complete. ∎

III. Proposed Solutions

We propose a framework, which tries to fix sensors' directions one-by-one in a greedy way depending on their "contributions" to coverage. We then propose two contribution functions when pointing a sensor to a particular direction. The first contribution function favors sensors making more total contributions to objects' coverage, while the second favors sensors adding more contributions to objects with higher coverage levels. The framework is outlined as follows:

1) Initially, we assume that all sensors' states are *undecided*.
2) For each undecided sensor s_i, we compute s_i's contribution when fixing its direction α_i to a particular angle in $[0, 2\pi)$, denoted by $contr(s_i, \alpha_i)$.
3) Let $contr(s_i, \alpha_i)$ be the largest contribution among all undecided s_i and its α_i. Then we point s_i toward α_i and change s_i's state to *fixed*.
4) Go back to step 2 to determine more sensors' directions, until any of the following conditions is met: (1) all sensors are fixed, (2) all objects are already (k, ω)-angle-covered, and (3) no undecided sensors can make any further contribution to any object's coverage level. If any condition is true, the algorithm is terminated and all remaining undecided sensors can be put to the sleep mode.

There are three issues to be addressed in the above procedure. First, the contribution function is yet to be defined. Second, each $\alpha_i \in [0, 2\pi)$ has infinite possibilities to be tested. We will show that only finite possibilities need to be evaluated. Third, the object's coverage level should be calculated. These will be addressed below.

A. Contribution Functions

We propose two ways to define the contribution function. Let $o_j.level$ be the current angle-coverage level of o_j contributed by those fixed sensors, and $o_j.level'$ be the new angle-coverage level of o_j after fixing s_i to direction α_i. The first contribution function simply sums up the increments of all objects' coverage levels

$$contr_1(s_i, \alpha_i) = \sum_{\forall o_j}(o_j.level' - o_j.level).$$

For the second contribution function, let r_k and r'_k be the numbers of objects that are (k, ω)- or more than (k, ω)-angle-covered before and after, respectively, s_i becomes fixed. Also, let r_j and r'_j, $j = 1, \ldots, k-1$, be the numbers of objects that are exactly (j, ω)-angle-covered before and after, respectively, s_i becomes fixed. The second contribution function is a vector of length k:

$$contr_2(s_i, \alpha_i) = [r'_k - r_k, r'_{k-1} - r_{k-1}, \ldots, r'_1 - r_1].$$

We use lexicographic ordering to compare two length-k vectors. We say that $[v'_k, v'_{k-1}, \ldots, v'_1] > [v_k, v_{k-1}, \ldots, v_1]$ if $v'_k > v_k$ or there is an integer $i < k$ such that $v'_k = v_k, \ldots, v'_{i+1} = v_{i+1}$, and $v'_i > v_i$.

Intuitively, $contr_1()$ simply adds all increments of coverage levels together to compare two sensors' contributions. On the contrary, $contr_2()$ gives higher priority to those that make objects closer to the goal of becoming (k, ω)-angle-covered. The latter could be more favorable when we are short of sensors.

B. Enumerating Sensors' Directions

Our scheme relies on computing each undecided sensor's contribution when fixing its direction in $[0, 2\pi)$. Below, we show that there are only a finite number of possibilities to be enumerated. Consider any s_i and α_i. Let O' be the set of objects such that each $o_j \in O'$ satisfies $dis(s_i, o_j) \le r_i$. For each $o_j \in O'$, consider the angle between $dir(\overrightarrow{s_i o_j})$ and the x axis. We sort objects in O', according to these angles, in an ascending order, into a list $o_1, o_2, \ldots, o_p$, where $p = |O'|$.

Then we enumerate the possible values of α_i as follows. Initially, we tune α_i to $dir(\overrightarrow{s_i o_1}) + \theta$. This is the largest possible α_i such that o_1 is angle-covered. Then we gradually enter an iterative process by gradually rotating α_i. In each iteration, let $o_x, o_{x+1}, \ldots, o_y$ be the sub-list of $o_1, o_2, \ldots, o_p$ that are currently angle-covered by s_i. Then we rotate α_i in the counterclockwise direction by an amount θ' such that the list of objects that are angle-covered by s_i becomes $o_{x+1}, o_{x+2}, \ldots, o_y$ or $o_x, o_{x+1}, \ldots, o_{y+1}$. It is not hard to see that

$$\theta' = min\{dir(\overrightarrow{s_i o_{x+1}} - (\alpha_i - \theta)), dir(\overrightarrow{s_i o_{y+1}} - (\alpha_i + \theta))\}.$$

Then we rotate α_i to $\alpha_i + \theta'$ and complete this iteration. This causes either o_x leaving or o_{y+1} entering s_i's current angle-covered list. The process is repeated until α_i returns to its initial direction, $dir(\overrightarrow{s_i o_1}) + \theta$.

Clearly, the above process has a finite number of iterations, and generates a finite number of possible α_is.

C. Computing Objects' Coverage Levels

In Section III-A, we need to compute the current angle-coverage level $o_j.level$ of o_j contributed by those fixed sensors. Here we show how to calculate this in polynomial time. We first present a structure for determining from a given set the optimal subset of sensors that can provide the highest angle-coverage level of an object.

Optimal Structure: Consider an object o_j and a set of sensors $C = \{s_{i_1}, s_{i_2}, \ldots, s_{i_p}\}$, in which each sensor angle-covers o_j. Since not every sensor in C can contribute to o_j's angle-coverage level, the purpose is to identify the exact and largest subset $C' \subseteq C$ which gives the highest angle-coverage level of o_j. Given an optimal C', let us assume, without loss of generality, that s_{i_1} is an element in C'. Then we rotate the vector $\overrightarrow{o_j s_{i_1}}$ in the counterclockwise direction using o_j as the center, until the first sensor in C' is encountered. Let this sensor be s_x. Apparently, $\angle s_x o_j s_{i_1} \geq \omega$. Similarly, we can rotate $\overrightarrow{o_j s_{i_1}}$ again in the same way, until the first sensor $s_y \in C$ is encountered such that $\angle s_y o_j s_{i_1} \geq \omega$. Since the search space C is a superset of C', we have $\angle s_y o_j s_{i_1} \leq \angle s_x o_j s_{i_1}$. Then the set $C'' = (C' - \{s_x\}) \cup \{s_y\}$ is also an optimal set leading to the (same) highest angle-coverage level of o_j.

Essentially, the above optimal structure indicates that if we know a sensor in C that could constitute an optimal solution to the angle-coverage determination problem, then from that sensor we can easily find the next sensor that can join into the solution, which is ensured to be optimal, by greedily scanning in the counterclockwise direction until the first sensor with an angle no less than ω from the previous sensor is found. Therefore, a simple scheme as follows is guaranteed to find an optimal solution.

1) For each sensor $s_x \in C$, let $C' = \{s_x\}$ and try to enlarge C' as follows.
   a) Repeatedly rotate the vector $\overrightarrow{o_j s_x}$ in the counterclockwise direction using o_j as the center. Whenever a new sensor is found whose angle from the previous sensor is at least ω, add it to C'. This is repeated until all sensors are exhausted or a sensor whose angle to the first sensor s_x is less than ω is encountered.
2) The above step will construct $|C|$ sets. The one with the largest cardinality is an optimal solution that leads to the highest angle-coverage level to o_j.

IV. Some Extensions

In this section, we present two simple extensions to our scheme. The first one tries to make our scheme a distributed protocol. The second one tries to further reduce the number of active nodes. In the distributed protocol, we assume that some of the nodes are instructed to start the protocol. On hearing any neighboring node starting this protocol (by hearing any packets discussed below), a node will also start this protocol. For ease of presentation, we also assume that all nodes' transmission ranges and sensing ranges are uniform, and the former is at least two times the latter. (This assumption can be relaxed if a node can collect its neighborhood information via multi-hop communication.) The following protocol is presented using any s_i as the subject.

1) Initially, s_i's state is *undecided*.
2) Then s_i collects objects that can be angle-covered by itself and communicates with its (one-hop) neighbors whose states are fixed and collect their current directions. From these information, s_i calculates its contribution function $contr(s_i, \alpha_i)$ when fixing its direction toward α_i. (Either $contr_1(s_i, \alpha_i)$ or $contr_2(s_i, \alpha_i)$ defined earlier can be plugged into the contribution function.)
3) Then s_i periodically broadcasts a BID$(s_i, \alpha, contr(s_i, \alpha_i))$ packet to its (one-hop) neighbors with a preset period for a preset interval of ΔT_{bid}.
4) For any other node which receives s_i's BID, if its state is undecided, it will also run the above step 2 to decide its contribution function. If its contribution is higher than s_i's, it will run step 3 to broadcast its BID; otherwise, it will keep silent for an interval of ΔT_{wait} and then re-run step 3.
5) For s_i, if it receives any BID with a higher contribution after sending out its own BID, it will keep silent for an interval of ΔT_{wait} and then re-run step 3. Otherwise, after ΔT_{bid}, it will fix its direction at α_i decided in step 2 and change its state to *fixed*.
6) Once becoming fixed, s_i will periodically broadcast a WIN(s_i, α) packet with a preset period for a preset interval of ΔT_{win}.

The second extension is based on the observation that when our scheme terminates, some objects may remain under $(k-1, \omega)$-angle-covered or over $(k+1, \omega)$-angle-covered, which makes some sensors redundant. This is because of the greedy nature of our scheme. If a sensor observes that all objects that are angle-covered by itself are still under $(k-1, \omega)$-angle-covered, it can simply go to sleep. Similarly, if a sensor observes that all objects that are angle-covered by itself are over $(k+1, \omega)$-angle-covered, it can go to sleep too. These rules apply to both our centralized scheme and distributed protocol.

V. Simulation Results

We have simulated an environment with randomly deployed sensors and objects. We look at performance as well as complexity. In order to make comparisons, we also implement an intuitive exhaustive search. This brute-force algorithm tries to enumerate all possible combinations and select the best one. Unless stated otherwise, the following discussions assume a sensing field of 500×500 m^2 with default values of $\theta = \frac{\pi}{6}$, $r = 25$ m, $k = 3$, and $\omega = \frac{\pi}{6}$. In each experiment, each case is run at least 100 times.

Given m objects, we first investigate the ratio of angle-covered objects versus the time complexity incurred

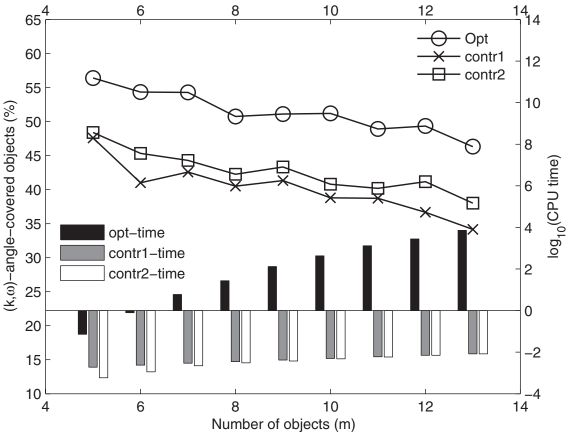


Fig. 3. Ratio of (k, ω)-angle-covered objects (in lines) and time complexity incurred (in bars) under different m.

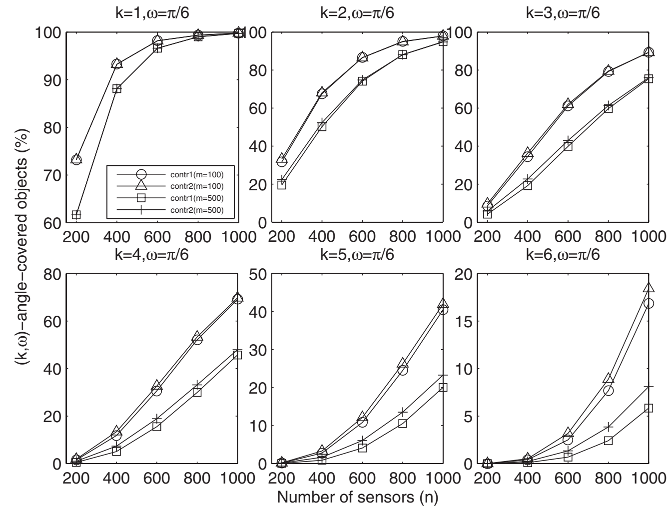


Fig. 4. Effect of n on the ratio of (k, ω)-angle-covered objects under different k.

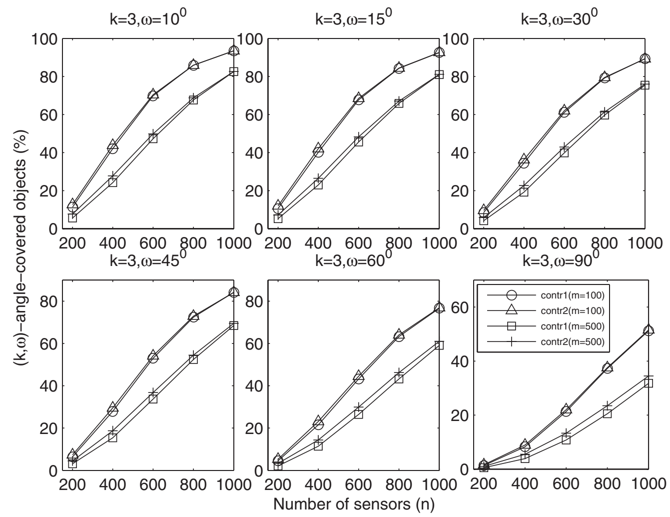


Fig. 5. Effect of n on the ratio of (k, ω)-angle-covered objects under different ω.

in each scheme. We assume $n = 10$ sensors and $k = 3$. Because the optimal solution has an exponentially increasing time complexity, we limit the sensing field to $100 \times 100\ m^2$ with $m = 5 \sim 20$ objects. Fig. 3 shows the ratio of objects that are (k, ω)-angle-covered and the CPU time incurred. This experiment is run on an Intel Q6600 2.4GHz CPU, with DDR2-800 4GB memory, and a WD SATA2 250GB hard disk. When $m \geq 12$, the CPU time of the optimal method becomes almost intolerable. The proposed methods incur relatively much less time. However, the ratios of (k, ω)-angle-covered objects of the optimal and the proposed methods are quite close (about $5 \sim 12\%$ less covered objects. To understand how our exhaustive search performs, first observe that its time complexity is mainly determined by three components: (1) sensors selection, (2) objects checking, and (3) angle checking. For simplicity, let each sensor be able to turn to d discrete directions. In the best case, the exhaustive search may luckily find that using k sensors is sufficient to (k, ω)-angle-cover all m objects. There are C_k^n ways to choose these sensors. There are d^k combinations to set these sensors' angles. For each setting, it takes $O(m)$ time to verify if all m are covered. So, the best-case time complexity is $O(C_k^n \cdot d^k \cdot m)$. In the worse case, there are $O(2^n)$ ways to select sensors. The number of angle combinations also raises to $O(d^n)$. So the worse-case time complexity is $O(2^n \cdot d^n \cdot m)$. Since the optimal method is computationally untractable, we will ignore it in the rest of our presentation.

In Fig. 4 and Fig. 5, we vary the number of sensors (n) to observe the ratio of (k, ω)-angle-covered objects under different k and ω, respectively. The number of objects (m) is fixed at 100 or 500. In Fig. 4, $\omega = \frac{\pi}{6}$ and $k = 1 \sim 6$. The performance of both contribution functions is quite close when m is smaller. However, $contr_2()$ is better than $contr_1()$ when m is larger. This is because $contr_2()$ works in a more greedy way. Also, when $k \geq 5$, the ratio of (k, ω)-angle-covered objects decreases quickly. It means that these sensors are insufficient to cover these objects. This is why we see more significant improvement when $k < 5$ if we increase n. When $k \geq 5$, increasing n has less impact. In Fig. 5, we set $k = 3$ and $\omega = 10° \sim 90°$. It is natural that a larger angle of separation ω causes a lower coverage ratio. This is especially true when $m = 500$. When $\omega \geq 60°$, the ratios are quite low because it is hard to find properly separated sensors, as shown in Fig. 5.

In Fig. 6 and Fig. 7, we investigate the effect of m on the ratio of covered objects under different k and ω, respectively. In Fig. 6, $\omega = \frac{\pi}{6}$ and $k = 1 \sim 6$. The coverage ratio highly depends on the density of sensors. The value of k also affects the coverage ratio; in particular, when $k > 3$, it drops significantly. Also, when k is larger, $contr_2()$ has clearer advantage over $contr_1()$. This again shows that $contr_2()$ is more favorable when we are short of sensors. In Fig. 7, we set $k = 3$ and $\omega = 10° \sim 90°$. Increasing ω generally degrades all schemes' performances. The impact is less significant when we change the value of m, but is more significant when we change the value of n. For example, the curve of $n = 1000$ degrades much quicker from that of $n = 500$.

In Fig. 8, we set $n = 1000$, $k = 3$, $\omega = \frac{\pi}{6}$, and vary θ from 15° to 75°. We can see that the effect of increasing θ is minor

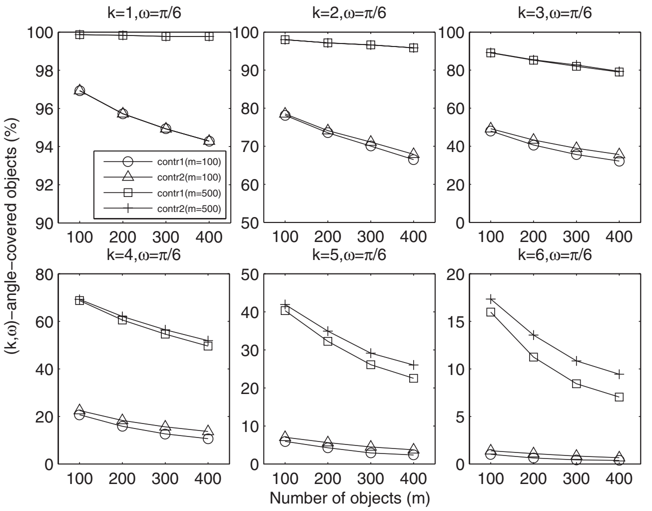


Fig. 6. Effect of m on the ratio of (k, ω)-angle-covered objects under different k.

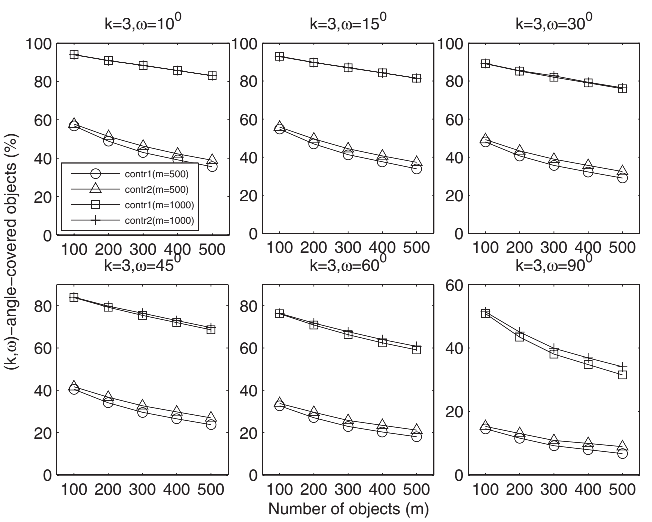


Fig. 7. Effect of m on the ratio of (k, ω)-angle-covered objects under different ω.

when the density of objects is low ($m = 100$), but is more significant when the density of objects is high ($m = 500$). This is reasonable because a larger θ helps sensors to cover more objects, especially when m is larger.

In Fig. 9, given $m = 400$ and 800 objects, we keep on increasing the number of sensors until all objects are covered. Initially, the curves rise fast and then are getting saturated after exceeding 80% of coverage. Since sensors are randomly added, after 80% coverage, adding more sensors has very limited effect for both $m = 400$ and 800 and both contribution functions.

In Fig. 10, using the same simulation parameters in Fig. 3, we compare the ratio of (k, ω)-angle-covered objects given by centralized and distributed methods under different m. The result shows that the coverage ratio of the distributed method loses 10%∼15% coverage as compared with the centralized method. Because a sensor only obtains its one-hop neighborhood information, this distributed solution solved only be adopted when objects are mobile. Function $contr_2(distributed)$ works slightly better than

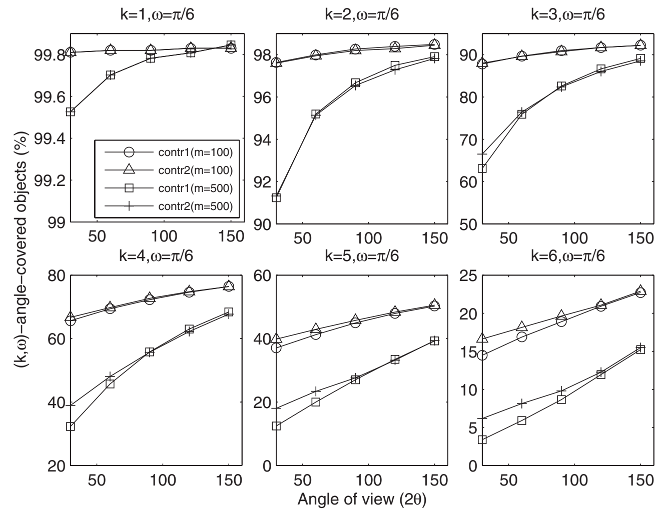


Fig. 8. Effect of angle θ on the ratio of (k, ω)-angle-covered objects under different k.

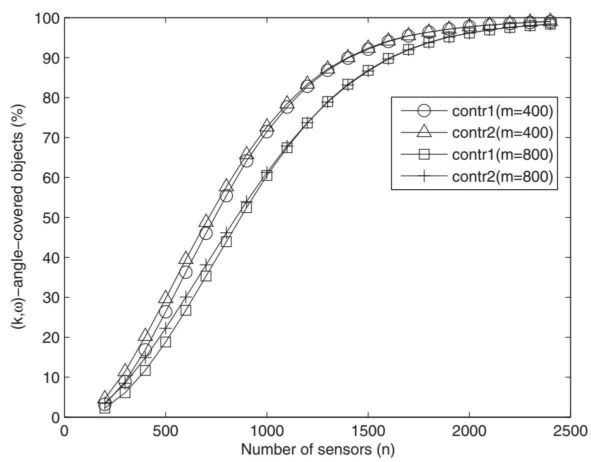


Fig. 9. Impact of adding more sensors for achieving 100% coverage.

$contr_1(distributed)$ when m is small, but the effect is more significant when m is larger.

We summarize three points as follows: (1) Function $contr_2()$ works slightly better than $contr_1()$ when sensors are relatively sparser than objects. (2) Adding sensors and increasing θ can improve the ratio of (k, ω)-angle-coverage objects. (3) Parameter k is also an impact factor on the number of needed sensors.

VI. Prototyping Experiences

We have developed a small-scale prototype to demonstrate the usefulness of angle coverage. The system architecture is shown in Fig. 11. The prototype consists of some cameras, some objects, some obstacles, and a monitoring server. To avoid the complicated object recognition job, we use some colored cubes as objects (to simulate human) and tape two toy eyes on one side of the cube (to simulate a human face). In our current prototype, four colors (blue, red, green, and yellow) are used to distinguish different objects. Note that if more objects are needed, we can use some unique color bars on each side of a cube to represent its identity. With these simplifications, we

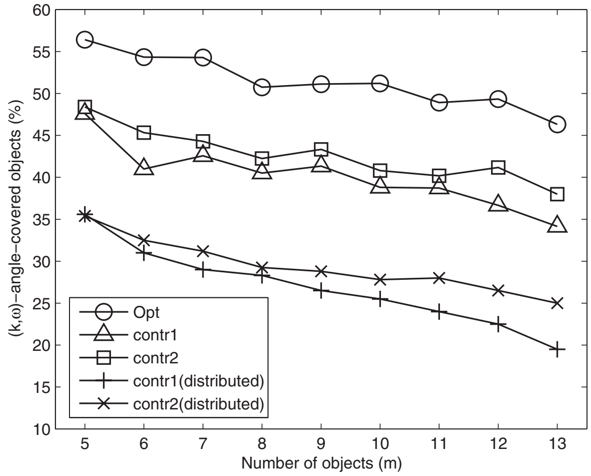


Fig. 10. Ratio of (k, ω)-angle-covered objects under different m.

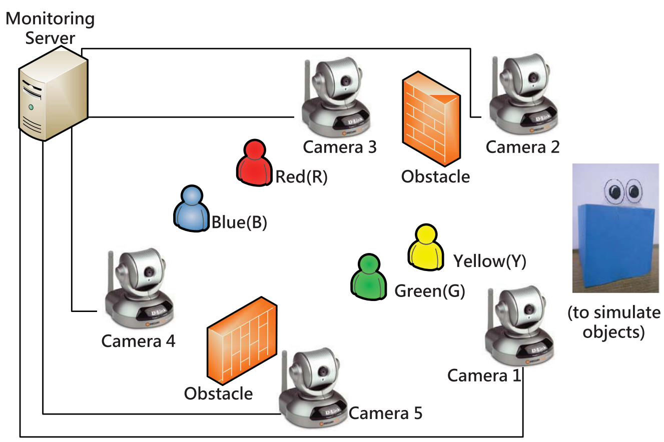


Fig. 11. Our prototyping system architecture.

try to develop a prototype for angle coverage. We adopt the *DCS-5220* wireless Internet camera made by D-Link Inc. with 4x digital zoom capability and pan and tilt functions that can cover 270 degrees horizontally and 90 degrees vertically. This camera has a 1/4" CMOS sensor and a standard 4 mm lens with 0.5 Lux @ F2.0. The video resolution is 30 fps with a frame size 640 × 480. Each camera has one default monitoring direction and eighteen rotatable angles. The configuration can be done by sending HTTP commands via wired or wireless links (here we adopt wired line). Besides, we use a white background to reduce the color distortion problem.

The monitoring server collects videos from cameras and runs the proposed contribution functions to orient each camera's direction. Our system uses colors to identify different objects. On receiving the video stream from a camera, the monitoring server uses the IBM Java Toolkit [35] to analyze it. The following procedure will be executed for each video stream received from each camera. The server first retrieves images from the video stream and then analyzes them based on the RGB color model to extract the RGB value of each pixel. Then these RGB values are converted into the *Hue*, *Saturation*, and *Value* (HSV) model, which is one of the most common cylindrical-coordinate representations of points in an RGB color model. We identify an object based on its *hue* value that is the main property of a color. (We do not use a fixed

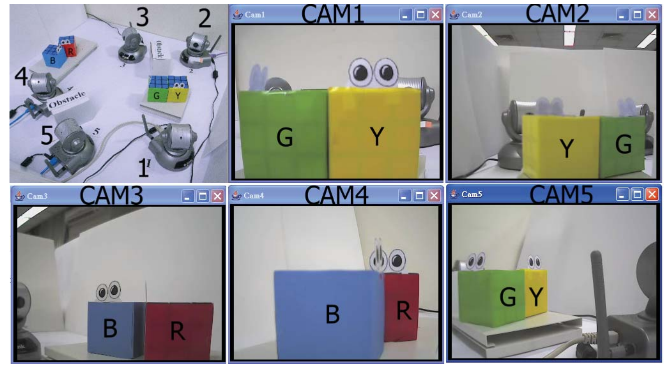


Fig. 12. Demonstration of using $contr_1()$ to control camera setting.

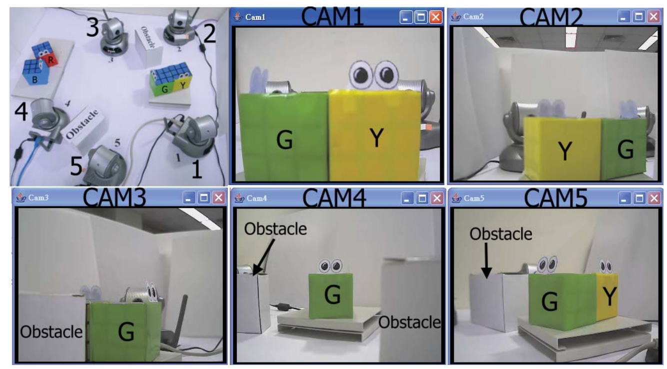


Fig. 13. Demonstration of using $contr_2()$ to control camera setting.

threshold since this will be too sensitive to light exposure.) In our system, we set a threshold=2000 on the number of color pixels to decide whether a camera can identify an object, i.e., if the number of color pixels of an image in a video frame exceeds 2000, the camera will treat it as an object. In other words, if an object is too far away from a camera, the camera will not consider this object.

The system execution steps are outlined as follows:

1) Initially, we set all cameras' default monitoring directions. Then we let cameras rotate their angles and report their captured video streams at different angles.
2) For each *undecided* camera s_i, we compute s_i's contribution on each particular angle α_i and record its value of $contr(s_i, \alpha_i)$ in the server. Note that we have to check the angle-constrain ω for s_i and other *decided* cameras covering the same objects while calculating s_i's contribution.
3) Let $contr(s_i, \alpha_i)$ be the largest contribution among all undecided s_i. We then point s_i toward α_i. (In some cases, we may get the same contribution value for different cameras when computing $contr_1()$. We break this tie by comparing the amounts of color pixels observed by them because more pixels mean that an object is closer to a camera.
4) Go back to step 2 to determine more cameras' directions, until all cameras are fixed.

Our experimental environment consists of five cameras, four objects, two obstacles, and one monitoring server and we set $(k, \omega) = (4, \frac{\pi}{4})$. Some interesting experimental results are shown in Fig. 12 and Fig. 13. Fig. 12 shows that using $contr_1()$

results in red and blue cubes being $(2, \frac{\pi}{4})$-angle-covered and green and yellow cubes being $(3, \frac{\pi}{4})$-angle-covered by our cameras. On the other hand, Fig. 13 shows that using $contr_2()$ only results in two objects, Green and Yellow, being $(5, \frac{\pi}{4})$- and $(3, \frac{\pi}{4})$-angle-covered by our cameras, respectively. The main differences between Fig. 12 and Fig. 13 are due to the limitations of cameras 3 and 4. Both cameras can choose to track either the Green cube or the Blue and the Red cubes simultaneously. Due to the property of $contr_1()$, cameras 3 and 4 will try to cover more objects. On the contrary, $contr_2()$ tries to achieve a high coverage level first. Also, in our previous system [36], we find that light is an uncertain factor that may drive into different experimental results, because light intensity does affect the RGB value of each pixel. For this reason, we use the HSV model to analyze video images instead of using the RGB model. The difficulty to identify colors by using RGB model is on setting a suitable threshold when the environmental light varies fast. In the HSV model, only the *hue* value is influenced by the light but it can be solved easily by setting its upper and lower bounds for RGB colors, respectively. So before starting the experiment, an initial process is needed to set the upper and lower bounds of *hue* value to filter out noise to increase the accuracy of identifying result. In our test, the upper and lower bound of the *hue* value of RGB colors are set to $(-4°, -12°)$, $(112°, 79°)$, $(238°, 149°)$, and $(81°, 45°)$ for red, green, blue, and yellow, respectively. The other values, the lower bounds of *saturation* and *value*, are not affected by the light intensity significantly, so they are usually tuned once. In our experiment, we set S(red=29, green=7, blue=17, yellow=24) and V(red=47, green=25, blue=34, yellow=56). In addition, the operation time, including the time caused by initial process (5∼10 seconds) and system execution system (∼30 seconds), needs around 35∼40 seconds. In the system execution, it needs two seconds to rotate a camera and capture video frames, from each angle.

VII. Conclusion

In this paper, we have defined a new k-angle object coverage problem in wireless sensor networks and proposed centralized and distributed methods based on two different contribution functions to solve this problem. The first contribution function fixes sensors that can add the largest overall contributions first while the second function fixes sensors that can add the largest numbers of higher angle-covered objects first. Extensive simulations have been conducted based on different parameters. Finally, we have built a prototype to demonstrate the feasibility of the proposed method applying to real applications. We believe that our work has built a fundamental basis for the angle-coverage-related research. Future works include studying the case with movable objects and prototyping more real applications.

References

[1] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw.*, 2001, pp. 221–235.

[2] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, pp. 1567–1576, Jun. 2002.

[3] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proc. ACM Int. Workshop Wireless Sensor Netw. Appl.*, 2002, pp. 22–31.

[4] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *ACM Commun.*, vol. 43, no. 5, pp. 51–58, May 2000.

[5] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE Int. Conf. Comput. Commun.*, pp. 775–784, Mar. 2000.

[6] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2001, pp. 166–179.

[7] Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," *Comput. J.*, vol. 47, no. 4, pp. 448–460, 2004.

[8] J. O'Rourke, *Art Gallery Theorems and Algorithms*. London, U.K.: Oxford Univ. Press, 1987.

[9] A. Heppes and J. B. M. Melissen, "Covering a rectangle with equal circles," *Period. Math. Hungarica*, vol. 34, nos. 1–2, pp. 65–81, 1996.

[10] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *ACM Mobile Netw. Appl.*, vol. 10, no. 4, pp. 519–528, 2005.

[11] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Netw.*, vol. 11, no. 3, pp. 333–340, 2005.

[12] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 189–200.

[13] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *Proc. IEEE Int. Conf. Comput. Commun.*, Mar.–Apr. 2003, pp. 1073–1083.

[14] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, 2003, pp. 28–39.

[15] C.-F. Huang, Y.-C. Tseng, and H.-L. Wu, "Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network," *ACM Trans. Sensor Netw.*, vol. 3, no. 1, p. 5, Mar. 2007.

[16] G. Tan, S. Jarvis, and A. Kermarrec, "Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 836–848, Jun. 2009.

[17] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proc. Int. Conf. Distrib. Comput. Syst.*, 2003, pp. 28–37.

[18] C.-F. Huang, L.-C. Lo, Y.-C. Tseng, and W.-T. Chen, "Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 182–187, May 2006.

[19] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Comput. Commun.*, vol. 29, no. 4, pp. 413–420, Feb. 2006.

[20] A. Boukerche and X. Fei, "A coverage-preserving scheme for wireless sensor network with irregular sensing range," *Ad Hoc Netw.*, vol. 5, no. 8, pp. 1303–1316, Nov. 2007.

[21] Y. Cai, W. Lou, M. Li, and X. Li, "Energy efficient target-oriented scheduling in directional sensor networks," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1259–1274, Sep. 2009.

[22] X. Cao, X. Jia, and G. Chen, "Maximizing lifetime of sensor surveillance systems with directional sensors," in *Proc. 6th IEEE Int. Conf. Mobile Ad-Hoc Sensor Netw.*, Dec. 2010, pp. 110–115.

[23] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2005, pp. 284–298.

[24] L. Zhang, J. Tang, and W. Zhang, "Strong barrier coverage with directional sensors," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 2009, pp. 1–6.

[25] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2005, pp. 1–7.

[26] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *J. Combinat. Optim.*, vol. 11, no. 1, pp. 21–41, Feb. 2006.

[27] E. Hörster and R. Lienhart, "On the optimal placement of multiple visual sensors," in *Proc. ACM Int. Workshop Video Surveill. Sensor Netw.*, 2006, pp. 111–120.

[28] G. Fusco and H. Gupta, "Selection and orientation of directional sensors for coverage maximization," in *Proc. IEEE Commun. Soc. Conf. Sensor Mesh Ad Hoc Commun. Netw.*, Jun. 2009, pp. 1–9.

[29] Y. Osais, M. St-Hilaire, and F. Yu, "Directional sensor placement with optimal sensing range, field of view and orientation," *Mobile Netw. Appl.*, vol. 15, no. 2, pp. 216–225, Apr. 2010.

[30] C.-W. Su, H.-Y. M. Liao, H.-R. Tyan, C.-W. Lin, D.-Y. Chen, and K.-C. Fan, "Motion flow-based video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1193–1201, Oct. 2007.

[31] J.-W. Hsieh, Y.-T. Hsu, H.-Y. M. Liao, and C.-C. Chen, "Video-based human movement analysis and its application to surveillance systems," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 372–384, Apr. 2008.

[32] W.-K. Leow, C.-C. Chiang, and Y.-P. Hung, "Localization and mapping of surveillance cameras in city map," in *Proc. ACM Int. Conf. Multimedia*, 2008, pp. 369–378.

[33] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *Proc. 3rd Annu. IEEE Commun. Soc. Sensor Ad Hoc Commun. Netw.*, Sep. 2006, pp. 374–382.

[34] D. Hochbaum, *Approximation Algorithms for NP-Hard Problems*. Boston, MA: PWS Publishing, 1996.

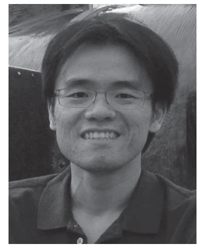
[35] *IBM Toolkit for MPEG-4 SDK* [Online]. Available: http://www.ibm.com

[36] P.-Y. Chen, H.-M. Lin, W.-T. Chen, and Y.-C. Tseng, "A multi-view visual surveillance system based on angle coverage," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2010, pp. 357–358.

Yu-Chee Tseng (S'91–M'95–SM'03–F'12) received the Ph.D. degree in computer and information science from Ohio State University, Columbus, in 1994.

He was with the Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, as a Chairman from 2005 to 2009, and as a Chair Professor and Dean since 2011. His current research interests include mobile computing, wireless communication, and sensor networks.

Dr. Tseng was the recipient of the Outstanding Research Award from the National Science Council, in 2001, 2003, and 2009, the Best Paper Award from the International Conference on Parallel Processing in 2003, the Elite I. T. Award in 2004, the Distinguished Alumnus Award from Ohio State University in 2005, and the Y. Z. Hsu Scientific Paper Award in 2009. He served on the editorial boards of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2005 to 2009 and the IEEE TRANSACTIONS ON MOBILE COMPUTING from 2006 to 2011, and has served on the editorial board of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS since 2008.



Po-Yu Chen (S'04–M'06) received the B.S. degree from the Department of Electrical Engineering, Chang Gung University, Taiwan, in 2001, and the M.S and Ph.D. degrees from the Institute of Communications Engineering, National Tsing Hua University, Taiwan, in 2003 and 2009, respectively.

He is currently a Research Fellow with the Department of Computer Science, National Tsing Hua University. He is also the Project Manager of the MediaTek Embedded Systems Technology Research and Personnel Training Program supported by MediaTek Inc. This project investigates important issues in the embedded system, such as power saving, system visualization, and biosign detection. His current research interests include wireless ad hoc and sensor networks, vehicular ad hoc networks, and LTE-A networks.

Wen-Tsuen Chen (M'87–SM'90–F'94) received the B.S. degree in nuclear engineering from National Tsing Hua University, Taiwan, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1970, 1973, and 1976, respectively.

He has been with the National Tsing Hua University since 1976 and as a Distinguished Chair Professor of the Department of Computer Science. He has served as Chairman of the Department, Dean of College of Electrical Engineering and Computer Science, and the President of National Tsing Hua University. Since March 2012, he has been with the Academia Sinica, Taiwan, as a Distinguished Research Fellow of the Institute of Information Science. His current research interests include computer networks, wireless sensor networks, mobile computing, and parallel algorithms.

Dr. Chen received numerous awards for his academic accomplishments in computer networking and parallel processing, including the Outstanding Research Award of the National Science Council, the Academic Award in Engineering from the Ministry of Education, the Technical Achievement Award and Taylor L. Booth Education Award of the IEEE Computer Society, and is currently a lifelong National Chair of the Ministry of Education, Taiwan. He is the Founding General Chair of the IEEE International Conference on Parallel and Distributed Systems, and the General Chair of the IEEE International Conference on Distributed Computing Systems among others in 2000. He is a fellow of the Chinese Technology Management Association.