Heterogeneous Directional Sensors Self-Deployment Problem in a Bounded Monitoring Area†

Charka Panditharathne, Ting-Yu Lin*, and Kuan-Hua Chen
Department of Communication Engineering
National Chiao-Tung University

Abstract— **Good deployment of sensors empowers the network with effective monitoring ability. Different from omnidirectional sensors, the coverage region of a directional sensor is determined by not only the sensing radius (distance), but also its sensing orientation and spread angle. Heterogeneous sensing distances and spread angles are likely to exist among directional sensors, to which we refer as heterogeneous directional sensors. In this paper, we target on a bounded monitoring area and deal with heterogeneous directional sensors equipped with locomotion and rotation facilities to enable the sensors self-deployment. Our base and optimized deployment algorithms are proposed to achieve high sensing coverage ratio in the monitored field. These algorithms leverage the concept of virtual forces (for sensors movements) and virtual boundary torques (for sensors rotations). Performance results demonstrate that our optimized deployment mechanism is capable of providing desirable surveillance level, while consuming moderate moving and rotating energy.**

Keywords— **Directional sensors deployment, sensing coverage, virtual boundary torques.**

I. Background

A wireless sensor network (WSN) is widely used for habitat and environmental surveillance, medical application (with the purpose of improving quality of health care), agricultural assistance, and as solutions to military problems [2, 5, 9, 10]. Wireless sensors generally come in two sensing shapes: omnidirectional (disk-shaped) and directional (fan-shaped). As illustrated in Table I, omnidirectional sensors obey the circular sensing model, while directional sensors have sector-like sensing behavior. An evident difference between the two types of sensors is that a directional sensor is identified by both its location (position) and sensing direction (orientation). Another difference in terms of sensing coverage is that, ***the covered region of a directional sensor is determined by not only the sensing radius, but also its sensing spread angle***. Consequently, an arbitrary directional sensor s_i is generally represented by a 5-tuple (x_i, y_i, r_i, θ_i, δ_i), where (x_i, y_i) denotes the coordinate (location) of s_i, r_i expresses the sensing radius, θ_i points out the viewing angle (sensing orientation), and δ_i indicates the sensing spread angle. In this paper, the viewing angle θ_i is defined as the angle this sensor is facing with respect to the horizontal line of $y = y_i$ in counterclockwise direction, where $0 \leq \theta_i < 2\pi$.

For a successful network surveillance (whether it is omnidirectional or directional WSN), providing sufficient sensing coverage is essential. Manual placement of static sensors involves labor effort and lacks network self-healing competence (when faulty sensors occur). Thanks to the availability of motion and rotation facilities, we consider smart sensors with movable and rotatable capabilities to accomplish self-deployment after an initial random placement of sensors. A number of research efforts have explored the movement-assisted sensors deployment techniques by utilizing mobile sensors to enhance the sensing coverage [12, 14, 15, 18]. However, those works target on omnidirectional sensors.

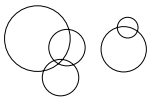
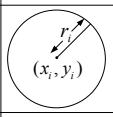
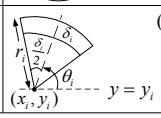
Due to the increasing popularity of surveillance camera networks and video sensing applications, several research works on the coverage problem of directional sensor networks have recently emerged [1, 3, 8, 13, 17]. In [3], the authors introduce a distributed algorithm that schedules the sensing directions (orientations) of active sensors to maximize the covered area. Also studying the coverage problem for directional sensors with tunable orientations, [1] proposes scheduling algorithms to minimize the set of active sensors while maximizing the number of monitored targets. Taking another perspective on the coverage problem, authors in [8] study how many sensors N are needed to meet a given required coverage probability p, in a directional sensor network where sensors are uniformly distributed. In addition, given fixed sensor population N, the problem of how sensing radius r should be adjusted is also discussed. However, the above works do not tackle the sensors deployment problem directly. To address the deployment problem, authors in [13, 17] propose coverage-aware algorithms for deploying directional sensors by rotating sensors toward coverage holes. All of the aforementioned research works deal with homogeneous directional sensors in their approaches. Nevertheless, heterogeneity in sensing radius and spread (offset) angle commonly exists among directional sensors. Thus deployment algorithms that consider heterogeneous directional sensors are practically necessary, despite the design challenges.

In this paper, we do *not* intend to address the issue of required amount of sensors for achieving certain degree of sensing coverage. Rather, given any number of sensors, we investigate the *heterogeneous sensors deployment problem* by proposing algorithms to improve the coverage ratio after a possibly random deployment of sensors. Tapping into the movable and rotatable capabilities enabled by locomotion and revolving facilities geared on sensors, we develop self-deployment algorithms with the goal of achieving high (desirable) coverage ratio in a bounded monitoring area. Several unique contributions are made in this work. Firstly, we observe that ***rotating directional sensors at fixed***

*Corresponding author (E-mail: ting@cm.nctu.edu.tw).
†This research was co-sponsored in part by the NSC of Taiwan under grant number 99-2221-E-009-006, and in part by the MoE Program Aiming for the Top University and Elite Research Center Development Plan (ATU Plan).

TABLE I

General classification of sensors based on circular (omnidirectional) or sector-like (directional) sensing shapes.

Attribute / Type	Sensing Shape	Sensing Capability	Sensor Representation
Type-I (disk-shaped)		temperature, humidity, brightness, etc.	 (x_i, y_i) : location of sensor s_i r_i : sensing radius of s_i
Type-II (fan-shaped)		camera, infrared, ultrasound, etc.	 (x_i, y_i) : location of sensor s_i r_i : sensing radius of s_i θ_i : viewing angle of s_i δ_i : spread angle of s_i

positions is not as effective as re-distributing them to adequate positions, in terms of extending sensing coverage. Instead of merely revolving sensors ([13, 17]), our deployment approaches translocate sensors with reasonable traveling distances. Rotations are applied each time after moderate movements have been performed to further improve the sensing coverage. Secondly, since heterogeneity of directional sensors includes both the different sensing distances and varying spread angles, simultaneously taking both parameters into account complicates the protocol design. Therefore we propose to create *virtual omnidirectional sensors* with circular sensing shape that approximates the original sensing sector. Different combinations of sensing distance and spread angle in directional sensors now map to distinct values of sensing radius in the created virtual omnidirectional sensors. In this way, ***we successfully transform the two-parameter deployment function into a single-parameter method***. Then our base deployment algorithms operate on those virtual omnidirectional sensors and guide the actual directional sensors to self-deploy themselves accordingly. Thirdly, ***a novel concept of virtual boundary torques is introduced to assist in sensors rotations***. The basic idea of exercising virtual boundary torques is to keep sensors staying inside the monitoring region and facing outward. Our optimized deployment algorithm leverages this concept and further enhances the coverage performance, while pleasantly conserves total energy consumption by preventing sensors from traveling far.

The remainder of this paper is organized as follows. In Section II, we summarize the environmental assumptions made in this work. Our base and optimized deployment algorithms are elaborated in Section III and Section IV, respectively. Section V presents the performance results, with respect to obtained sensing coverage and total energy consumed by physical sensors movements and rotations. Finally, we draw our conclusion in Section VI.

II. Environmental Assumptions

In this paper, we focus on the deployment problem in a bounded monitoring area, which can be generally approximated as a rectangular region. The boundaries do *not* refer to physical walls that prohibit sensors from moving outside the monitored zone. Rather, the boundaries define *an area of interest*, and sensors do not necessarily always stay inside the monitored zone after performing the self-deployment algorithms. However, the sensing coverage outside the monitored zone is wasted since we are not interested. Therefore in the proposed algorithm, we aim to reduce such wasted coverage and increase the effective *sensing coverage ratio*, which is defined as the percentage of area covered by at least one sensor in the bounded monitoring region. Below we summarize our environmental assumptions.

(A1) There exists a powerful clusterhead responsible for performing centralized computations. All sensors are able to communicate with the clusterhead via single-hop or multi-hop wireless transmissions.

(A2) Sensors have sector-like sensing shape and the binary sensing/detection behavior, in which an event is detected (not detected) by a sensor with complete certainty if this event occurs inside (outside) its sensing coverage. Both the homogeneous and heterogeneous sensors are allowed in our model. Information of respective sensing distances and spread angles is provided by all sensors and made available at the clusterhead for deployment-related computations.

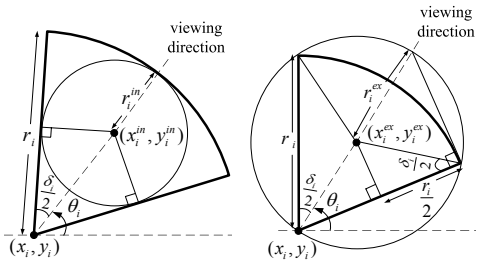
(A3) We adopt the discrete coordination system, in which the monitoring area (sensing field) is represented by a 2D grid network. Locations and orientations of all sensors are obtained via the pre-deployed RFID platform or some existing localization technique combined with compass system, and constantly updated to the clusterhead. Neighboring nodes under the adopted coordination system are defined as sensors within the sensing range (r_s), which is normally much smaller than the radio communication distance (r_c). Although the sensing behavior is directional, we have the omnidirectional communication model in our network. Without loss of generality, we assume that $r_c > 2r_s$ in our model. According to the derivations in [7, 16], if the radio communication range (r_c) is at least twice the sensing radius (r_s), complete coverage of a convex area implies connectivity among the working set of sensor nodes. Consequently, in this work, we only deal with the sensing coverage, and network connectivity follows accordingly.

III. Base Deployment Algorithms

We propose to create virtual omnidirectional sensors that simulate the actual directional sensors. By mapping distinct combinations of sensing distance and spread angle in directional sensors to dissimilar values of sensing radius in the created virtual omnidirectional sensors, we reduce the originally two-parameter deployment function into a single-parameter operation. Consequently, simple yet effective al-

gorithms can be devised for deploying heterogeneous directional sensors. Two ways of obtaining the corresponding virtual omnidirectional sensors are presented in Section III-A. Then Section III-B details how virtual forces are applied to those virtual sensors for increasing sensing region covered by the actual sensors.

A. Creating Virtual Omnidirectional Sensors



(a) (b)

Fig. 1. Obtaining (a) the inscribed circle and (b) the circumscribed circle of a fan-shaped sensor.

To approximate a sector (fan), we obtain the inscribed circle (internally tangent circle) or the circumscribed circle, as illustrated in Fig. 1. For a directional sensor s_i located at (x_i, y_i) with sensing distance r_i and spread angle δ_i, the obtained inscribed circle (abbreviated as *incircle*) indicates the sensing region of the corresponding virtual omnidirectional sensor s_i^{in} located at (x_i^{in}, y_i^{in}) with sensing radius r_i^{in}. Through simple geometric calculations, we have

$$\begin{aligned} x_i^{in} &= x_i + (r_i - r_i^{in})\cos\theta_i \\ y_i^{in} &= y_i + (r_i - r_i^{in})\sin\theta_i, \end{aligned} \tag{1}$$

where $r_i^{in} = \frac{r_i \sin(\frac{\delta_i}{2})}{1+\sin(\frac{\delta_i}{2})}$. Likewise, the obtained circumscribed circle (shortened as *excircle*) renders the sensing region of the corresponding virtual omnidirectional sensor s_i^{ex} located at (x_i^{ex}, y_i^{ex}) having sensing radius r_i^{ex}. Based on similar geometric computations, we have

$$\begin{aligned} x_i^{ex} &= x_i + r_i^{ex}\cos\theta_i \\ y_i^{ex} &= y_i + r_i^{ex}\sin\theta_i, \end{aligned} \tag{2}$$

where $r_i^{ex} = \frac{r_i}{2\cos(\frac{\delta_i}{2})}$. As a result, we construct a *virtual sensors set* $\{s_1', s_2', \ldots, s_k'\}$ for the original sensors set $\{s_1, s_2, \ldots, s_k\}$, given k sensors available for our deployment. Based on whether incircles or excircles are created, the virtual sensors set $\{s_1', s_2', \ldots, s_k'\}$ can be made $\{s_1^{in}, s_2^{in}, \ldots, s_k^{in}\}$ or $\{s_1^{ex}, s_2^{ex}, \ldots, s_k^{ex}\}$. Our virtual forces calculations operate on the virtual sensors set directly. Next we describe the designs and actions of virtual forces.

B. Applying Virtual Forces

The concept of virtual forces is inspired by the combined idea of potential field and disk packing theory [4, 6]. Each sensor behaves as a source giving a force to others. This force can be either positive (attractive) or negative (repulsive). If two sensors are too close, they exert repulsive forces to separate each other, otherwise they exert attractive forces to draw each other. We quantify the definition of "closeness" by using the distance threshold d_{th}^{ij} for any two sensors s_i and s_j with respective sensing radius r_i and r_j. On the other hand, define parameters w_a and w_r as the weight measurements associated with the attractive and repulsive forces separately. The designs of d_{th}^{ij} and weight constants (w_a, w_r) are critical for the deployment efficacy. Due to space limitation, we provide the design details in [11]. According to our derivations, the distance threshold should be set on a node-pair basis, and thus $d_{th}^{ij} = \alpha(r_i + r_j)$, where $0 < \alpha \leq 1$ (in our simulations, we set $\alpha = 0.86$). For the weight constants, we prove that good choices for w_a and w_r greatly depend on sensor population and monitored area dimensions, while independent of sensing radius. Thus, given k omnidirectional sensors to be deployed in an $m \times n$ monitoring area, the settings should be $w_a = \frac{1}{k}$ and $w_r = k\sqrt{m^2+n^2}$. Following the parameter design guidelines suggested in [11], we then apply the virtual forces algorithm on those virtual omnidirectional sensors created in Section III-A. Algorithm 1 expresses the pseudocode for our base deployment strategy. Depending on whether incircles or excircles are obtained, the created virtual sensors set $\{s_1', s_2', \ldots, s_k'\}$ can be $\{s_1^{in}, s_2^{in}, \ldots, s_k^{in}\}$ or $\{s_1^{ex}, s_2^{ex}, \ldots, s_k^{ex}\}$. In the former case, we refer to the deployment approach as InCircle Algorithm (ICA), while in the latter case, we entitle it as ExCircle Algorithm (ECA).

Algorithm 1 Base Deployment Algorithm

```
create corresponding virtual omnidirectional sensors {s_1', s_2', ..., s_k'}
    for all directional sensors {s_1, s_2, ..., s_k};
set loops = 0;
set c_now = c_init;  // initial coverage ratio
while (loops < Maxloops) && (c_now < c_th) do
    for each virtual omnidirectional sensor s_i' ∈ {s_1', s_2', ..., s_k'} do
        compute F_i'=Σ_{j≠i,j=1}^{k} F_ij' + F_ib';
    end for
    perform virtual movements on all sensors;
    update coverage ratio c_now;
    set loops = loops + 1;
end while
```

The algorithm terminates when either the required sensing coverage threshold (c_{th}) or the maximum allowable number of iterations ($Maxloops$) is reached. Note that in the end of each loop (iteration), every sensor performs *virtual movement* without physically moving to the new position. Physical movements are conducted once the base deployment computation process terminates.

IV. Optimized Deployment Algorithm

After re-positioning sensors, we observe that the sensing coverage can be further enhanced by swiveling sensors to face toward boundaries. While the use of virtual forces can guide sensors to move, it is incapable of manipulating sensors viewing (facing) angles. Therefore, in Section IV-A, we introduce a novel concept of virtual torques that are given by the boundaries to guide sensors rotations. Then an optimized deployment algorithm that combines virtual forces and boundary torques is proposed in Section IV-B.

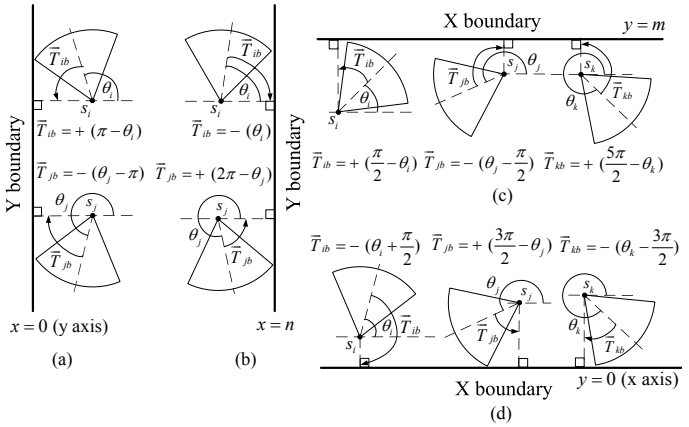


Fig. 2. Computations of virtual boundary torques for boundaries along (a) y axis, (b) $x = n$, (c) $y = m$, and (d) x axis, respectively in a 2D bounded monitoring area.

A. New Concept of Virtual Boundary Torques

A torque for sensor s_i is basically the amount of rotating angle centered at location (x_i, y_i). The rotating angle can be counterclockwise or clockwise. In this paper, we define the counterclockwise rotation as positive (+) and the clockwise negative (−). Virtual boundary torques are virtual torques received by a sensor from the boundaries. When there is only one boundary, the virtual torque is given such that a sensor can alter its viewing direction toward the boundary. In our coordinate system, illustrated in Fig. 2, there are two types of Y boundaries, including $x = 0$ and $x = n$, and two kinds of X boundaries, containing $y = 0$ and $y = m$. Below we discuss how to model the virtual boundary torques under the four conditions.

Define the virtual boundary torque given to sensor s_i as $\overrightarrow{T}_{ib}$. When we consider the boundary of $x = 0$, as shown in Fig. 2(a), the virtual boundary torque $\overrightarrow{T}_{ib}$ for sensor s_i with viewing angle θ_i no greater than π can be obtained as $+(\pi - \theta_i)$, which directs s_i to revolve counterclockwise for $(\pi - \theta_i)$ degrees. On the other hand, for sensor s_j with viewing angle θ_j greater than π, the virtual boundary torque $\overrightarrow{T}_{jb}$ can be computed as $-(\theta_j - \pi)$, indicating s_j should rotate clockwise for $(\theta_j - \pi)$ degrees. The purpose of separating the two cases is to limit each rotation angle within π (half circle), so that rotations energy can be reasonably conserved. Similar design principles are applied to other boundary conditions. For the boundary of $x = n$, two cases are considered, while three cases should be modeled separately for both X boundaries ($y = 0$ and $y = m$). Below we summarize all cases for computing the virtual boundary torques.

For sensor s_i having the boundary of $x = 0$ (y axis), as shown in Fig. 2(a), we have

$$\overrightarrow{T}_{ib} = \left\{ \begin{array}{ll} +\ (\pi - \theta_i) & \text{if } 0 \le \theta_i \le \pi \\ -\ (\theta_i - \pi) & \text{otherwise} \end{array} \right\}. \quad (3)$$

For sensor s_i having the boundary of $x = n$, as shown in Fig. 2(b), the virtual boundary torque $\overrightarrow{T}_{ib}$ is modeled as

$$\overrightarrow{T}_{ib} = \left\{ \begin{array}{ll} -\ (\theta_i) & \text{if } 0 \le \theta_i \le \pi \\ +\ (2\pi - \theta_i) & \text{otherwise} \end{array} \right\}. \quad (4)$$

For sensor s_i having the boundary of $y = m$, as shown in Fig. 2(c), we have virtual boundary torque

$$\overrightarrow{T}_{ib} = \left\{ \begin{array}{ll} +\ (\frac{\pi}{2} - \theta_i) & \text{if } 0 \le \theta_i \le \frac{\pi}{2} \\ -\ (\theta_i - \frac{\pi}{2}) & \text{if } \frac{\pi}{2} < \theta_i \le \frac{3\pi}{2} \\ +\ (\frac{5\pi}{2} - \theta_i) & \text{otherwise} \end{array} \right\}. \quad (5)$$

For sensor s_i having the boundary of $y = 0$ (x axis), as shown in Fig. 2(d), we obtain the virtual boundary torque $\overrightarrow{T}_{ib}$ as

$$\overrightarrow{T}_{ib} = \left\{ \begin{array}{ll} -\ (\theta_i + \frac{\pi}{2}) & \text{if } 0 \le \theta_i \le \frac{\pi}{2} \\ +\ (\frac{3\pi}{2} - \theta_i) & \text{if } \frac{\pi}{2} < \theta_i \le \frac{3\pi}{2} \\ -\ (\theta_i - \frac{3\pi}{2}) & \text{otherwise} \end{array} \right\}. \quad (6)$$

Next, we extend the concept to consider virtual torques from multiple boundaries. In a 2D rectangular region, we define the virtual boundary torque imposed on a sensor as *a combined torque from the nearest X boundary and nearest Y boundary*. Depending on the coordinate of sensor s_i, our Q function firstly determines which quadrant s_i resides in, where

$$Q(x_i, y_i) = \left\{ \begin{array}{ll} \text{I} & \text{if } x_i \le \frac{n}{2} \ \&\&\ y_i > \frac{m}{2} \\ \text{II} & \text{if } x_i > \frac{n}{2} \ \&\&\ y_i > \frac{m}{2} \\ \text{III} & \text{if } x_i \le \frac{n}{2} \ \&\&\ y_i \le \frac{m}{2} \\ \text{IV} & \text{if } x_i > \frac{n}{2} \ \&\&\ y_i \le \frac{m}{2} \end{array} \right\}. \quad (7)$$

As illustrated in Fig. 3(a), for sensors located in quadrant I, the boundaries of $x = 0$ and $y = m$ need be considered. Similarly, for sensors falling in quadrants II, III, and IV, one nearest X boundary and one nearest Y boundary are considered to produce a combined virtual boundary torque. Take sensor s_i in Fig. 3(a) for example, denote the virtual torque from X boundary as $\overrightarrow{T}^x_{ib}$ and torque from Y boundary as $\overrightarrow{T}^y_{ib}$, the resultant virtual boundary torque exerted on s_i is a combined torque of $\overrightarrow{T}^x_{ib}$ and $\overrightarrow{T}^y_{ib}$. Instead of combining the two torques with equal importance, we define a weight parameter $\beta = \frac{d^x_{ib}}{d^y_{ib}}$, where d^x_{ib} and d^y_{ib} denote the Euclidean distances between s_i and the nearest X and Y boundaries respectively. The weight parameter β is then used to give the closer boundary more impact on the resultant torque computation as we define

$$\overrightarrow{T}_{ib} = \frac{\overrightarrow{T}^x_{ib} + \beta \overrightarrow{T}^y_{ib}}{1 + \beta}. \quad (8)$$

Once the resultant virtual torques are obtained, sensors perform virtual rotations, as illustrated in Fig. 3(b). Some of the sensors in this figure may seem to render wasted coverage outside the boundaries after rotations. Nevertheless, in our optimized deployment algorithm (presented in the next subsection), an iteration actually includes both the exertions of virtual forces and virtual boundary torques. Therefore, those sensors have good chance to be moved inward in the following iterations.

B. Optimized Deployment Adapting to Sensor Population

As presented in Section III, the virtual omnidirectional sensors can be obtained as incircles or excircles. We observe that by using incircles, more overlappings are produced among the actual sensors. On the flip side, excircles allow

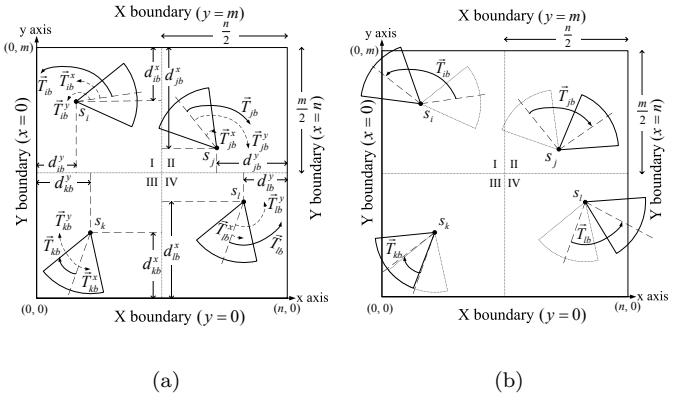


Fig. 3. Illustration of (a) resultant virtual boundary torques and (b) virtual rotations applied to respective sensor node.

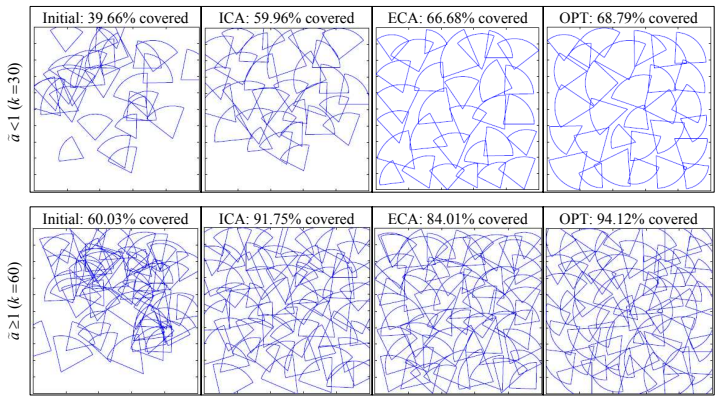


Fig. 4. Sensors deployment status using ICA, ECA, and OPT strategies, respectively ($m = 300, n = 300$, and HRA settings).

more spacings between sensors. This motivates us to devise an adaptive deployment algorithm that adapts to actual sensor population. When the available sensors are sufficient to fully cover the area, we prefer overlappings between sensors to reduce surveillance holes. In contrast, when sensors are insufficient, more spacings may be beneficial. Given the monitoring area of size $A = m \times n$, and the total area covered by all sensors $A_s = \sum_{i=1}^{k} \pi r_i^2 (\frac{\theta_i}{2\pi})$, define a factor $\widetilde{a} = \frac{A_s}{A}$, which indicates the maximal possible coverage ratio. We propose to create incircles when $\widetilde{a} \geq 1$, and use excircles otherwise.

Algorithm 2 provides the pseudocode for our optimized deployment mechanism. Note that the forces calculations operate on virtual sensors, while boundary torques computations work on actual sensors. Both virtual movements and rotations exert on actual sensors, but without physically moving and rotating themselves. Physical movements and rotations are conducted once the deployment calculation process terminates.

Algorithm 2 Optimized Deployment Algorithm (OPT)

1: **if** $\widetilde{a} < 1$ **then**
2: obtain *virtual omnidirectional sensors* $\{s_1', s_2', ..., s_k'\}$ as $\{s_1^{ex}, s_2^{ex}, ..., s_k^{ex}\}$; // excircles
3: **else**
4: obtain *virtual omnidirectional sensors* $\{s_1', s_2', ..., s_k'\}$ as $\{s_1^{in}, s_2^{in}, ..., s_k^{in}\}$; // incircles
5: **end if**
6: set $loops = 0$;
7: set $c_{now} = c_{init}$; // initial coverage ratio
8: **while** $(loops < Maxloops)$ && $(c_{now} < c_{th})$ **do**
9: **for** each virtual sensor $s_i' \in \{s_1', s_2', ..., s_k'\}$ **do**
10: compute $\overrightarrow{F}'_i = \sum_{j \neq i, j=1}^{k} \overrightarrow{F}'_{ij} + \overrightarrow{F}'_{ib}$;
11: **end for**
12: perform *virtual movements* on all sensors;
13: **for** each actual sensor $s_i \in \{s_1, s_2, ..., s_k\}$ **do**
14: compute virtual boundary torque $\overrightarrow{T}_{ib}$;
15: **end for**
16: perform *virtual rotations* on all sensors;
17: update *coverage ratio* c_{now};
18: set $loops = loops + 1$;
19: **end while**

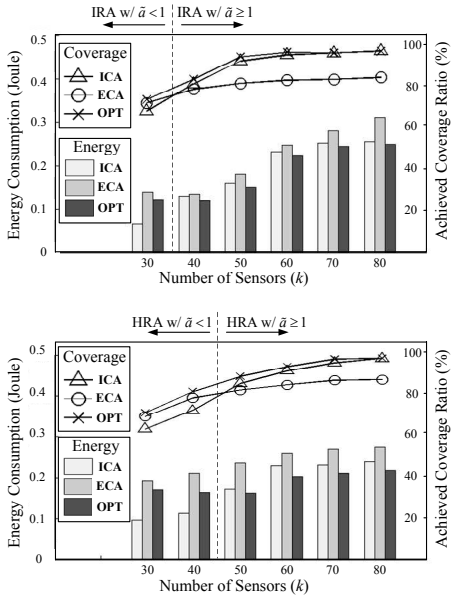


Fig. 5. Coverage ratios attained and energy consumed by ICA, ECA, and OPT mechanisms under various amounts of sensor nodes in a monitored 300×300 area with IRA and HRA settings.

V. Performance Evaluation

Since only rotations are performed, the deployment mechanisms proposed by [13,17] do not perform well in our simulated environments. In addition, their mechanisms consider only homogeneous directional sensors. To the best of our knowledge, no previous deployment algorithms that mesh sensor movements with rotations are available for our comparison. Thus, in this section, we analyze the performance yielded by the proposed base and optimized (enhanced) deployment algorithms in terms of achieved sensing coverage and consumed energy from physical movements and rotations. We simulate both homogeneous and heterogeneous directional sensors. Homogeneous sensors have identical sensing radius at 66 and spread angle at $\frac{2\pi}{5}$. For heterogeneous sensors, we set the sensing radius uniformly distributed in $[45, 75]$ and spread angle uniformly distributed in $[\frac{\pi}{3}, \frac{\pi}{2}]$. We use abbreviation IRA to indicate the setting for homogeneous sensors with Identical Sensing Radius

and Spread Angle, and HRA to imply the setting for heterogeneous sensors with Heterogeneous Sensing Radius and Spread Angle. Three proposed deployment algorithms are implemented: ICA, ECA, and OPT. All three algorithms use $Maxloops = 1000$ and $c_{th} = 0.98$ as their termination conditions.

Fig. 4 displays the deployment results produced by ICA, ECA, and OPT mechanisms with 30 and 60 heterogeneous directional sensors. Our OPT strategy performs like ECA when $\widetilde{a} < 1$ ($k = 30$), and similar to ICA when $\widetilde{a} \geq 1$ ($k = 60$). Nonetheless, since OPT also performs torques calculations (for virtual rotations) right after forces computations (for virtual movements) in each iteration, it is capable of achieving a even higher sensing coverage than the other two strategies. In the generated scenarios, the effect of sensors rotations is not as pronounced as that of sensors movements. By wisely combining the two actions (movements and rotations) in the deployment function, OPT effectively improves the sensing coverage ratio by 73% (when $k = 30$) and 57% (when $k = 60$) respectively.

In Fig. 5, we observe the obtained coverage ratio and energy consumption required by respective deployment strategy. The energy consumed by OPT includes both the physical moving and rotating energy, while ICA and ECA only consume physical moving energy. To model the energy, we do real experiments on the sensor robot used in our prototyping testbed with grid size equal to 1 cm. The robot assembles six 1.2 V 2000 mAh rechargeable NiMH batteries with measured $200 \sim 290$ mA moving current and average moving speed at 0.06 m/sec (216 m/hr). Consequently, the average moving energy consumption per grid (unit distance) can be estimated at $0.29 \times 7.2 \times (\frac{0.01}{216}) = 9.667 \times 10^{-5}$ Joule. On the other hand, $280-310$ mA rotating current and average rotating speed at 96 degree/sec (345600 degree/hr) are measured on the same robot device. Thus energy consumption per degree can be estimated at $0.31 \times 7.2 \times (\frac{1}{345600}) = 6.458 \times 10^{-6}$ Joule. As shown in Fig. 5, since OPT has the adaptiveness to sensor population, higher sensing coverage can always be achieved no matter how many sensors are available for deployment. An interesting result exhibited in this figure is that OPT does not consume the most energy among the three strategies, despite having the best sensing coverage. This phenomenon is because, by guiding sensors facing outward, OPT actually consumes less moving energy than ICA and ECA. The saved moving energy sufficiently makes up for the rotating energy spent by OPT. Hence we demonstrate that our OPT deployment mechanism can be both coverage-effective and energy-conserving.

VI. Conclusion

In this paper, we propose effective self-deployment algorithms for heterogeneous directional sensing sensors. Our results demonstrate that by judiciously guiding sensors movements and rotations, the critical sensing coverage ratio can be enhanced without incurring significant energy cost.

References

[1] J. Ai and A. A. Abouzeid. "Coverage by Directional Sensors in Randomly Deployed Wireless Sensor Networks". *Journal of Combinatorial Optimization*, 11(1):21–41, February 2006.

[2] E. S. Biagioni and K. W. Bridges. "The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species". *Int'l Journal of High Performance Computing Applications*, 16(3):315–324, 2002.

[3] W. Cheng, S. Li, X. Liao, S. Changxiang, and H. Chen. "Maximal Coverage Scheduling in Randomly Deployed Directional Sensor Networks". *In Proc. Int'l Conf. Parallel Processing Workshops (ICPPW)*, September 2007.

[4] A. Howard, M. J. Mataric, and G. S. Sukhatme. "Mobile Sensor Network Deployment Using Potential Fields: A Distributed Scalable Solution to the Area Coverage Problem". *In Proc. Int'l Symposium on Distributed Autonomous Robotics Systems*, June 2002.

[5] E. Jovanov, D. Raskovic, J. Price, J. Chapman, A. Moore, and A. Krishnamurthy. "Patient Monitoring Using Personal Area Networks of Wireless Intelligent Sensors". *Biomedical Sciences Instrumentation*, 37:373–378, 2001.

[6] M. Locatelli and U. Raber. "Packing Equal Circles in a Square: A Deterministic Global Optimization Approach". *Elsevier Discrete Applied Mathematics*, 122(1-3):139–166, October 2002.

[7] J. Lu and T. Suda. "Differentiated Surveillance for Static and Random Mobile Sensor Networks". *IEEE Transactions on Wireless Communications*, 7(11):4411–4423, November 2008.

[8] H. Ma and Y. Liu. "Some Problems of Directional Sensor Networks". *Int'l Journal of Sensor Networks*, 2(1-2):44–52, April 2007.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. "Wireless Sensor Networks for Habitat Monitoring". *In Proc. Int'l Workshop on Wireless Sensor Networks and Applications (WSNA)*, September 2002.

[10] S. A. Musman, P. E. Lehner, and C. Elsaesser. "Sensor Planning for Elusive Targets". *Elsevier Mathematical and Computer Modelling*, 25(3):103–115, February 1997.

[11] C. Panditharathne, T.-Y. Lin, and W.-T. Liu. "Enhanced Virtual Forces Algorithm with Boundary Forces". *Technical Report (available at http://bunlab.twbbs.org/filezone/files/EVFA-B.pdf)*, 2010.

[12] K.-H. Peng, T.-Y. Lin, and K.-H. Chen. "On Seamless Wireless Sensor Deployment and System Implementation". *Int'l Journal of Advanced Information Technologies (IJAIT)*, 3(2):72–92, December 2009.

[13] D. Tao, H. Ma, and L. Liu. "Coverage-Enhancing Algorithm for Directional Sensor Networks". *Lecture Notes in Computer Science (LNCS)*, 4325:256–267, November 2006.

[14] G. Wang, G. Cao, and Thomas F. La Porta. "Movement-assisted Sensor Deployment". *IEEE Transactions on Mobile Computing*, 5(6):640–652, June 2006.

[15] J. Wu and S. Yang. "SMART: A Scan-based Movement-assisted Sensor Deployment Method in Wireless Sensor Networks". *In Proc. IEEE INFOCOM*, March 2005.

[16] H. Zhang and J. C. Hou. "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks". *Ad Hoc and Sensor Wireless Networks*, 1(1-2):89–124, March 2005.

[17] J. Zhao and J.-C. Zeng. "A Virtual Potential Field Based Coverage Algorithm for Directional Networks". *In Proc. Chinese Control and Decision Conference (CCDC)*, pages 4590–4595, 2009.

[18] Y. Zou and K. Chakrabarty. "Sensor Deployment and Target Localization Based on Virtual Forces". *In Proc. IEEE INFOCOM*, April 2003.