Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Autonomous deployment of wireless sensor networks for optimal coverage with directional sensing model☆

Feng Li [a,*], Jun Luo [b], Shiqing Xin [c], Ying He [b]

[a] *School of Computer Science and Technology, Shandong University, Jinan, China*
[b] *School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore*
[c] *Institute of Computer Science and Technology, Ningbo University, Ningbo, China*

ABSTRACT

Sensor deployment is an important aspect of network architecture for *Wireless Sensor Networks* (WSNs). Although many solutions to mobile sensors deployment have been proposed, controlling mobile sensors with directional sensing ability towards optimal coverage remains to be an open problem. In this paper, we take the initiative to handle the *Coverage Maximizing Mobile Sensor Deployment Problem* (CMMSDP) with directional and arbitrarily oriented sensors. Our proposal consists of two algorithms. The first one, *Concurrent Rotation and Motion Control* (CRMC), is a localized iterative algorithm derived from optimality conditions, so it aims at reaching local maximum. The second algorithm, *Staged Rotation and Motion Control* (SRMC), decouples rotation and motion controls in order to reduce the computation complexity with slight sacrifice in optimality. We derive optimality and complexity results for both algorithms. We also implement our algorithms in TOSSIM and evaluate them against commonly used metrics. The promising results confirm the absolute feasibility of our proposals.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

The deployment of sensor nodes is very crucial to the functionalities of *Wireless Sensor Networks* (WSNs) [1]. Specifically, a deployment should guarantee *area of interest* is largely covered. However, a well arranged deployment, on one hand, leads to a prohibitive cost due to the large scale of a WSN, and on the other hand, is lack of adaptability to the monitoring of time-variant events, which has forced many of WSN related proposals to resort to random deployments. Although random deployments may satisfy certain coverage requirements given an over-provisioned node density, the incurred cost is huge, while the lack of adaptability to event dynamics remains to be an inevitable issue.

To tackle these challenges, we adopt sensor nodes that are equipped with mobility mechanisms, e.g., wheels driven by DC motors, compass and bumper [2,3]. As sensor nodes can move to-wards desirable locations, the initial random deployments can be improved by properly adjusting the locations of the nodes [4–6]. Such an autonomous deployment strategy with the help of mobile sensor nodes enables flexible re-deployment when the physical phenomena of interest vary during surveillance, or the network condition changes (e.g., in the face of node failures). However, the existing proposals on autonomous deployment only consider sensor nodes with omnidirectional sensors and boolean sensing ranges, hence a disk centered at each node is used to characterize the coverage of that node [4,5,7,8]. As real sensors may have certain directional features (e.g., radar or acoustic sensors [9,10]) and the sensing capability of a sensor often decreases continuously with an increasing distance from the sensor rather than remaining constant but suddenly becoming zero somewhere [11], specific deployment mechanism is expected.

In this paper, we take the initiative to deal with the *Coverage Maximizing Mobile Sensor Deployment Problem* (CMMSDP), under the assumptions of directional sensors with a general sensitivity distribution function. We aim at moving nodes to maximize coverage, which is indicated by an objective function. For omnidirectional sensors, this optimization problem is partially handled by *Centroidal Voronoi Tessellations* (CVT) [11–13]. However, directional sensors with arbitrary orientations lead to a Voronoi tessellation consisting of non-convex and even disconnected (in topological sense) cells with curved boundaries. This has made it highly

nontrivial to obtain CVT through a localized control algorithm, and the optimality of CVT may also become questionable.

In response to these challenges, we propose two autonomous deployment algorithms to control the node movements and orientations using only local information, based on a characterization of the optimal solutions to CMMSDP. Our first algorithm is CRMC (*Concurrent Rotation and Motion Control*), it simultaneously tunes the orientations and locations of nodes through a localized iteration, so that it terminates with a local maximum coverage. In order to improve the algorithm efficiency, our second algorithm SRMC (*Staged Rotation and Motion Control*) first unifies the orientations of the sensor nodes and then conducts the motions of the nodes. As a result, SRMC is more efficient in terms of computations. To the best of our knowledge, our proposal is the first to handle CMMSDP with directional and arbitrarily oriented sensors.

The remaining of this paper is organized as follows. We discuss existing literature in Section 2. Then we present our model and define our problems in Section 3, and analyze the theoretical characterization of the optimal solution in Section 4. We present CRMC and SRMC in Sections 5 and 6, respectively. We evaluate the performance of our algorithms and compare them with another possible solution in Section 7. Finally, we conclude our paper in Section 8.

2. Related work

There is a vast body of recent work related to sensor deployments for area coverage, including joint coverage and connectivity solutions to static sensor deployments (e.g., [14–16]) and intermittent coverage with mobile nodes (e.g. [17–19]). However, we focus only on those about deploying WSNs for better **constant coverage** using mobile nodes.

2.1. Moving node for improving coverage

This set of proposals aim at mobile deployment to only achieve better coverage, assuming (implicitly or explicitly) that the transmission range is sufficiently large so that coverage implies connectivity. Howard et al., [20] are among the first to devise the *virtual force approach* for mobile deployment. The idea is motivated by the attributes of electromagnetic particles: they push each other away when too close, while attracting each other when too far. This idea was extended by later proposals to combat its oscillatory behavior [5]. Wang et al., [4] pioneered in applying Voronoi diagrams to control mobility. They propose two methods, VOR and Minimax: while the former moves a node towards the farthest Voronoi vertex, the latter makes a node stop at the *Chebyshev center* (the center of the smallest circumscribed circle of the Voronoi cell). The experiments in [4] show that Voronoi-based approaches often perform better than the virtual force approach. CVT has recently been extended in [21] by jointly optimizing coverage and (the sensor nodes') moving distances in each iteration for the purpose of energy efficiency. Similarly, we also design our algorithm to carefully determine the step sizes of each iteration, aiming to ensure coverage optimality as well as affordable energy cost in moving nodes (see Section 5.3).

Whereas the aforementioned proposals always assume a circular sensing area with identical radius for all node, Bartolini et al., [5] apply Voronoi-Laguerre geometry [22] to deal with heterogenous sensing radius. Under heterogeneous sensing ranges, the optimality of Voronoi partition for fixed sensor locations [23] is compromised under Euclidean metric. Fortunately, by redefining the distance as *power distance* [24] (hence leading to Voronoi diagram in Laguerre geometry [22]), Bartolini et al., are able to retain certain properties of Voronoi diagram, and they hence reuse the Minimax method [4] to control mobility. Our later proposal is similar to [5], in the sense that we also exploit directional features of sensing

model such that classical theories/algorithms for standard Voronoi diagrams can be applied.

Different from the above proposals assuming omnidirectional sensors (hence circular sensing areas), [25] proposes mobility control algorithms to address the area coverage problem under a sector-based directional sensing model. It sticks to boolean sensing range and has the orientations of the directional sensors fixed in the deployment strategy; thereby lacking of practicality and optimality. In contrast, we present in this paper a more practical and generic sensing model, such that we can take into account both the continuity and directionality of sensing ranges, while guaranteeing the resolvability and optimality of our proposed solutions to the area coverage problem.

2.2. Integrated coverage and connectivity

When transmission range is roughly in the same scale of sensing range, the mobility control algorithm has to take network connectivity into account. However, as connectivity is a global property of a WSN, it cannot be reliably maintained by a localized algorithm without introducing redundancy (hence sacrificing coverage) [26]. As a result, the existing proposals all rely on certain global coordination mechanism and/or a geographic lattice known to all nodes (e.g., [7,27]). In particular, an underlying spanning topology (a backbone or tree) is constructed to maintain a global connectivity with, for example, a sink [27]. In addition, a coordinate system and a related lattice structure (hexagonal lattice for [27] and parallel lines for [7]) is maintained and is globally known; this helps to gradually "grow" a regular node deployment.

Although using a spanning topology is meaningful as the whole WSN always needs to connect to a sink, the requirement on a global coordinate system demands an expensive localization system that is not always possible. As our solution leads to an almost regular node deployment, connectivity is guaranteed by properly choosing a node density during the initial random deployment. Otherwise a spanning topology can be maintained along with our mobility control algorithms. Therefore, we do not involve connectivity in our optimization framework, though we still deal with the problems resulting from a limited transmission range.

3. Model, problem, and theory

We first present our system model in Section 3.1. We then review the basic theories for optimal deployment of omnidirectional sensors in Section 3.2, before formally formulating our optimization problem for maximizing coverage using directional mobile sensors in Section 3.3.

3.1. System model

We assume a WSN consisting of a set $\mathcal{N} = \{n_1, \ldots, n_N\}$ of sensor nodes, and $|\mathcal{N}| = N$. The nodes are initially deployed arbitrarily on a 2D targeted area $\mathcal{A}$.[1] We also make the following assumptions on sensor nodes:

A1: Each node is equipped with certain mechanisms (e.g., motors plus wheels) to gradually changes its location [2] as well as bumper sensors to detect and avoid obstacles in the targeted area [3].

A2: We hereby focus on a more practical sensing model where the sensing capability of a sensor is attenuated continuously with increasing distance [11] and the attenuation is supposed to be anisotropic [28] (Section 5.2.4: Variogram models, p70). Therefore, to characterize such a sensing model,

---

[1] Our solution is readily extensible to 3D surface or volume in theory, but our scope is restricted to 2D planes in this paper.

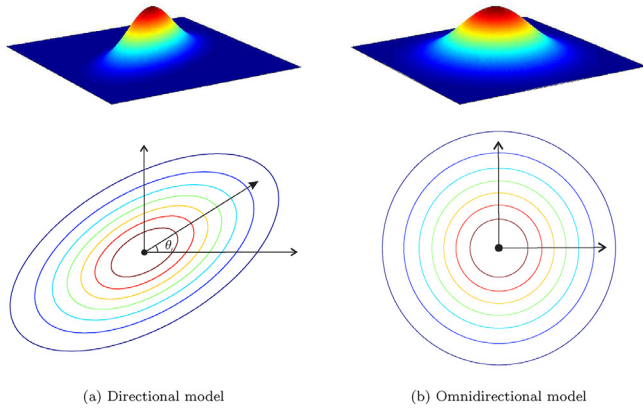(a) Directional model   (b) Omnidirectional model

Fig. 1. Two sensing models for sensor nodes.

we define a *Sensitivity Distribution Function* (SDF) for the ith sensor at location $u_i \in \mathcal{A}$ as $f(\|v - u_i\|^2_{P_i})$, where $v \in \mathcal{A}$, $\|v - u_i\|^2_{P_i} = (v - u_i)^T P_i^T P_i (v - u_i)$, and

$$P_i = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}, \qquad (1)$$

with $a, b > 0$ and $\theta_i \in [0, \pi)$ being the sensor *orientation*. In fact, $\|v - u_i\|^2_{P_i}$ can be treated as the distance from u_i to v under a directional metric P_i. We assume that $f(\cdot)$ is a **non-increasing** function with respect to $\|v - u_i\|^2_{P_i}$, to imply the attenuation of the sensing capability of node n_i according to increasing distance. Note that, we only require a SDF $f(\cdot)$ to be non-increasing, so it also includes the boolean-range sensing model as a special case, where $f(\cdot)$ is an *indicator function* of the sensing range. We illustrate a SDF in Fig. 1(a). If $a = b$, $P_i^T P_i = a^2 I$ where I is the identity matrix. The sensor becomes omnidirectional, as shown by Fig. 1(b). It can be viewed as a generalization of continuous omnidirectional sensing model with both directional feature and analytical tractability especially for our area coverage problem.

A3: Nodes all have an identical transmission range r, and we denote by $\mathcal{N}(n_i)$ the nodes within the transmission range of node n_i, i.e., $\mathcal{N}(n_i)$ is the set of *one-hop neighbors* of node n_i.

A4: A node n_i has a *gyroscope* to get aware of its orientation θ_i. It can use ranging information of $\mathcal{N}(n_i)$ to construct a local coordinate system for mobility control (e.g. [29]).

3.2. Preliminary theories for omnidirectional sensors

As mentioned above, the omnidirectional sensing model actually can be treated as a special case of our directional one. Therefore, before diving into our problem formulation based on our directional sensing model, we first review some basic theories closely related to omnidirectional sensors.

With omnidirectional sensors, we do not have to take into account $\{\theta_i\}$, such that the deployment problem actually is simpler than the case of directional ones. In order to allocate sensors to properly cover the given targeted region $\mathcal{A}$, the basic thread of deploying omnidirectional sensors is first to partition $\mathcal{A}$ into disjoint area $\{\mathcal{A}_i\}_{i=1,...,N}$ such that $\mathcal{A} = \bigcup_i \mathcal{A}_i$ and $\mathcal{A}_i \bigcap \mathcal{A}_j = \emptyset$ (with $i \neq j$), and then to allocate each sensor node n_i to one of the areas. We assign an *Event Density Function* (EDF) $\psi : \mathcal{A} \to \mathbb{R}_+$ upon the targeted region $\mathcal{A}$; it indicates the importance of different parts in $\mathcal{A}$. According to our definition of SDF, our aim is to maximize the fol-

lowing *coverage function*

$$\mathfrak{C}_{omni}(\{\mathcal{A}_i\}, \{u_i\}) = \sum_{i=1}^{N} \int_{\mathcal{A}_i} f(\|v - u_i\|^2_{\ell_2}) \psi(v) dv$$

where $\|\cdot\|^2_{\ell_2}$ denotes Euclidean norm as $a = b$ for individual sensor nodes in this case.

The optimal solutions thus can be characterized by the following two propositions [12,13].

Proposition 1. *If we fix the sensor locations $\{u_i\}$, the optimal partition of $\mathcal{A}$ is the Voronoi partition $\mathfrak{V}(\mathcal{A}) = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$ generated by $\{u_i\}$,*

$$\mathcal{V}_i = \left\{ v \in \mathcal{A} | \|v - u_i\|^2_{\ell_2} \leq \|v - u_j\|^2_{\ell_2}, \forall j \neq i \right\}.$$

While this proposition states the optimality of Voronoi partition given fixed sensor locations, the next one pinpoints the best Voronoi partitions, under rather restrictive conditions.

Proposition 2. *Let $f(x) = -x$, and define the centroid $C_\mathcal{V}$ of a region $\mathcal{V} \in \mathbb{R}^d$ $(d \geq 2)$ as $\left[\int_\mathcal{V} \psi(v) dv\right]^{-1} \left[\int_\mathcal{V} v \psi(v) dv\right]$ then $\mathfrak{C}_{omni}(\{\mathcal{V}_i\}, \{u_i\})$ is maximized **only if** $\{\mathcal{V}_i\}$ is generated by $\{u_i = C_{\mathcal{V}_i}\}$ for $i = 1, \ldots, N$.*

These two properties motivate the well known Lloyd's iteration and its gradient-based extensions [12], which basically repeat the two steps of (i) Voronoi partition generated by $\{u_i\}$ and (ii) moving $\{u_i\}$ to (or towards with a fractional step size) $\{C_{\mathcal{V}_i}\}$. We illustrate the initial and optimal deployments in Fig. 2. However, as the optimality requires $f(x) = -x$, we need to tackle this issue (even for omnidirectional sensors) in order to apply general SDFs.

3.3. Problem formulation

We hereby formulate our *Coverage Maximizing Mobile Sensor Deployment Problem* (CMMSDP) under our directional sensing model. Similar with the above case of handling omnidirectional sensors, we partition the targeted region $\mathcal{A}$ into a set of N disjoint areas $\{\mathcal{A}_i\}_{i=1,...,N}$. We allocate node n_i into $\mathcal{A}_i$ (tune its location $u_i \in \mathcal{A}_i$ and orientation θ_i) and let it take care of sensing $\mathcal{A}_i$. We define the coverage function $\mathfrak{C}(\cdot)$ for directional sensing model as

$$\mathfrak{C}(\{\mathcal{A}_i\}, \{u_i\}, \{\theta_i\}) = \sum_{i=1}^{N} \int_{\mathcal{A}_i} f(\|v - u_i\|^2_{P_i}) \psi(v) dv, \qquad (2)$$

hence our CMMSDP is given by

$$\underset{\{\mathcal{A}_i\}, \{u_i\}, \{\theta_i\}}{\text{maximize}} \quad \mathfrak{C}(\{\mathcal{A}_i\}, \{u_i\}, \{\theta_i\}) \qquad (3)$$

$$\text{subject to} \quad \bigcup_i \mathcal{A}_i = \mathcal{A}; \quad \mathcal{A}_i \bigcap \mathcal{A}_j = \emptyset; \quad u_i \in \mathcal{A}_i \qquad (4)$$

The aim is to determine the variables $\{\mathcal{A}_i\}$, $\{u_i\}$ and $\{\theta_i\}$ such that the coverage to the targeted area is maximized.

Although our CMMSDP falls into the so-called *location optimization framework* [23], the introduction of sensor orientations θ_i results in a new distance measure $\|v - u_i\|^2_{P_i}$ to replace the commonly used Euclidean metric, which makes our problem fairly unique and useful, especially considering the resulting sensing model is more generic and thus can convey better sensing characteristics than omnidirectional or binary sensing models. Nevertheless, as we will show in later sections, the new variables $\{\theta_i\}$ also lead to considerable challenges, which we have to address when designing specific algorithms.

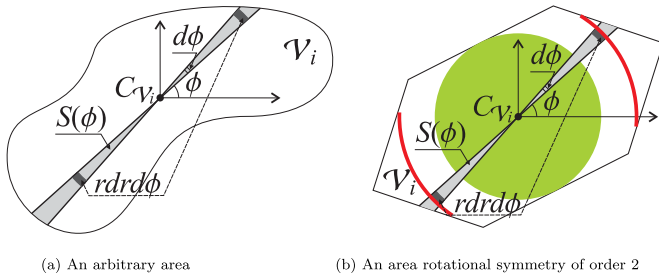## 4. The optimal solutions for directional sensors

Inspired by the theories of designing optimal deployment solutions for omnidirectional sensors, we present characterizations for
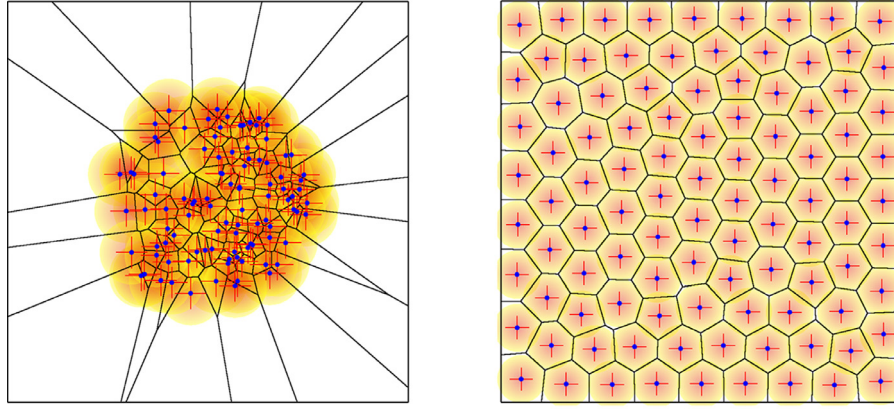
(a) Initial deployment   (b) Optimal deployment

Fig. 2. Voronoi cells for omnidirectional sensors.

(a) An arbitrary area   (b) An area rotational symmetry of order 2

Fig. 3. For an arbitrary area, the centroid $C_{\mathcal{V}_i}$ does not guarantee a zero inner integration (a). However, if an area is rotational symmetry of order 2, the inner integration equals zero for any $S(\phi)$ (b). Therefore, if the symmetry is preserved by additional operator (e.g., $f(\cdot)$ or P_i), the centroid remains intact under these operations.

directional sensors in this section, in order to motivate our theoretical results and algorithms for CMMSDP. As the objective function of CMMSDP $\mathfrak{C}(\{\mathcal{A}_i\}, \{u_i\}, \{\theta_i\})$ is generally not concave even with omnidirectional sensors, we are interested in deriving a local maximum which is sufficient for practical use.

4.1. Arbitrary SDFs

We first show that, under certain conditions, the centroids $\{C_{\mathcal{V}_i}\}$ also maximize $\mathfrak{C}(\{\mathcal{V}_i\}, \{u_i\})$ with an arbitrary $f(x)$.

Proposition 3. *If, for $i = 1, \ldots, N$, $\mathcal{V}_i$ is rotational symmetry of order 2 and the centroid $C_{\mathcal{V}_i}$ is the rotocenter, $\mathfrak{C}(\{\mathcal{V}_i\}, \{u_i\})$ is maximized for an arbitrary $f(x)$ **only if** $\{\mathcal{V}_i\}$ is generated by $\{u_i\}$ and $u_i = C_{\mathcal{V}_i}, \forall i = 1, \ldots, N$.*

Proof. The necessary condition for $\{u_i^*\}$ to maximize $\mathfrak{C}(\{\mathcal{V}_i\}, \{u_i\})$ is

$$\int_{\mathcal{V}_i} f'\left(\|v - u_i\|_{\ell^2}^2\right)(v - u_i)\psi(v)dv \bigg|_{u_i = u_i^*} = 0, \forall i = 1, \ldots, N.$$

We also know that $\int_{\mathcal{V}_i}(v - C_{\mathcal{V}_i})\psi(v)dv = 0$ or $\int_0^\pi \int_{S(\phi)} v_{r,\phi}\psi(v_{r,\phi})rdrd\phi = 0$ under polar coordinates centered at $C_{\mathcal{V}_i}$, where $S(\phi)$ is a pair of sectors in $\mathcal{V}_i$ between $[\phi, \phi + d\phi)$ and $[\pi + \phi, \pi + \phi + d\phi)$, and $v_{r,\phi}$ contains both $[r\cos\phi, r\sin\phi]$ and $[r\cos(\pi + \phi), r\sin(\pi + \phi)]$. We illustrate the latter integration under polar coordinates in Fig. 3(a). When $\mathcal{V}_i$ is rotational symmetry of order 2 and $C_{\mathcal{V}_i}$ is at the rotocenter, we have (under polar

coordinates centered at $C_{\mathcal{V}_i}$)

$$\int_{S(\phi)} f'(\|v_{r,\phi}\|_{\ell^2}^2)v_{r,\phi}\psi(v_{r,\phi})rdr = 0,$$

for all $i = 1, \ldots, N$, which is roughly explained by Fig. 3(b). Therefore, $\{u_i\} = \{C_{\mathcal{V}_i}\}$ is the necessary condition for maximizing $\mathfrak{C}(\{\mathcal{V}_i\}, \{u_i\})$. □

Recall that, for **omnidirectional sensors**, the outcome of Lloyd's iteration (and its variances) leads to almost symmetric Voronoi cells [12,13], especially when $\psi(v)$ is a constant (as assumed in [4]), as shown by Fig. 2 (b). These suggest that the conditions required by Proposition 3 are almost always satisfied, thus Lloyd's iteration does lead to local maximum or nearly maximum even if the cells are slightly out of symmetry.

4.2. Optimality conditions

We are now ready to characterize the optimal solutions of CMMSDP under directional sensing model, or equivalently the new metric $\|v - u_i\|_{P_i}^2$. Firstly, we have the optimality statement for Voronoi partition under fixed sensor locations and orientations, similar to Proposition 1.

Proposition 4. *If we fix the sensor locations $\{u_i\}_{i=1,\ldots,N}$ and orientations $\{\theta_i\}_{i=1,\ldots,N}$, the optimal partition of $\mathcal{A}$ is the Voronoi partition $\mathfrak{V}(\mathcal{A}) = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$ generated by $\{u_i\}_{i=1,\ldots,N}$ under $\{P_i\}_{i=1,\ldots,N}$, i.e.,*

$$\mathcal{V}_i = \left\{v \in \mathcal{A} \big| \|v - u_i\|_{P_i}^2 \le \|v - u_j\|_{P_j}^2, \forall j \ne i\right\}.$$

We omit the proof as it immediately follows from that $f(\cdot)$ is non-increasing in $\|v - u_i\|_{P_i}^2$. We also have the counterpart of Proposition 2 as follow, indicating the best Voronoi partitions under a specific SDF.

Proposition 5. *Assume $\mathcal{V}_i$ is rotational symmetry of order 2 and the centroid $C_{\mathcal{V}_i}$ is the rotocenter, then the objective function $\mathfrak{C}(\{\mathcal{V}_i\}, \{u_i\}, \{\theta_i\})$ is maximized **only if** $\{\mathcal{V}_i\}$ is generated by $\{u_i = C_{\mathcal{V}_i}\}$ under $\{P_i\}$, for $\forall i = 1, \ldots, N$.*

Proof. The necessary condition for $\{u_i^*\}$ to maximize $\mathfrak{C}(\{\mathcal{V}_i\}, \{u_i\}, \{\theta_i\})$ is

$$\int_{\mathcal{V}_i} f'\left(\|v - u_i\|_{P_i}^2\right)P_i(v - u_i)\psi(v)dv \bigg|_{u_i = u_i^*} = 0, \forall i = 1, \ldots, N.$$

We also have the centroidal condition $\int_{\mathcal{V}_i}(v - C_{\mathcal{V}_i})\psi(v)dv = 0, \forall i = 1, \ldots, N$. Following a similar reasoning of the previous proof (for

Proposition 3) and that P_i preserves rotational symmetry of even orders, we can show that the $\{u_i\} = \{C_{\mathcal{V}_i}\}$ implies the necessary optimality condition if $\mathcal{V}_i$ is rotational symmetry of order 2 and $C_{\mathcal{V}_i}$ is at the rotocenter. □

This proposition differs from Proposition 2 in that it does not assume a specific $f(\cdot)$ but demands a particular geometric property of $\mathcal{V}_i$.

The above results give us two optimality conditions: the optimal partition with fixed node locations and orientations, as well as the optimal node locations with fixed partition and orientations. What are the optimal orientations given an optimal Voronoi partition and optimal sensor locations?

Proposition 6. *If we fix the sensor locations $\{u_i\}$ and the Voronoi partition $\{\mathcal{V}_i\}$ generated by $\{u_i\}$ such that $u_i = C_{\mathcal{V}_i}$, then $\mathfrak{C}(\{\mathcal{V}_i\}, \{C_{\mathcal{V}_i}\}, \{\theta_i\})$ is maximized **only if***

$$\int_{\mathcal{V}_i} f'\left(\|v - C_{\mathcal{V}_i}\|_{P_i}^2\right) \frac{\partial \|v - C_{\mathcal{V}_i}\|_{P_i}^2}{\partial \theta_i} \psi(v) dv \Bigg|_{\theta_i = \theta_i^*} = 0, \quad (5)$$

for $i = 1, \ldots, N$, where $\|v - C_{\mathcal{V}_i}\|_{P_i}^2$ is a scalar function of θ_i.

The proof is omitted as it is just the first-order optimality condition with respect to θ_i. However, this optimality condition demands a careful interpretation, as it is far less straightforward as the other two. If we combine $f'(\|v - C_{\mathcal{V}_i}\|_{P_i}^2)$ and $\psi(v)$ to form a *virtual* EDF of the mass $\mathcal{V}_i$, the optimality condition states that the centroid of $\mathcal{V}_i$ is preserved under transform P_i. Moreover, obtaining θ_i^* can be further simplified under certain geometric property of $\mathcal{V}_i$.

Definition 1. For a mass $\mathcal{V}_i$ with an EDF ψ defined upon, if $\mathcal{V}_i$ is rotational symmetry of order 2 and the centroid $C_{\mathcal{V}_i}$ is the rotocenter, then we term w_i the *main axis* of $\mathcal{V}_i$ if

$$w_i = \arg\max_{\|w\|_{\ell 2} = 1} \int_{\mathcal{V}_i} (w^T (v - C_{\mathcal{V}_i}))^2 \psi(v) dv, \quad (6)$$

and we denote by $\beta_i \in [0, \pi)$ the orientation of w_i.

Proposition 7. *If $\mathcal{V}_i$ is rotational symmetry of order 2 and the centroid $C_{\mathcal{V}_i}$ is the rotocenter, $\theta_i^* = \beta_i, \forall i = 1, \ldots, N$.*

Proof. Changing the integration to polar coordinates centered at $C_{\mathcal{V}_i}$, we have

$$\int_{\mathcal{V}_i} f'\left(\|v - C_{\mathcal{V}_i}\|_{P_i}^2\right) \frac{\partial \|v - C_{\mathcal{V}_i}\|_{P_i}^2}{\partial \theta_i} \psi(v) dv$$
$$= \int_r r^3 \int_0^{2\pi} f'\left(\|v_{r,\phi}\|_{P_i}^2\right) g(\phi, \theta_i) \psi(v_{r,\phi}) dr d\phi, \quad (7)$$

where $v_{r,\phi} = [r\cos\phi, r\sin\phi]$, and $g(\phi, \theta_i) = (a^2 - b^2) \sin 2(\phi - \theta_i)$. We also redefine the main axis under the polar coordinates centered at $C_{\mathcal{V}_i}$ as follows,

$$\beta_i = \arg\max_{\theta_w} \int_r r^3 \int_0^{2\pi} \cos^2(\phi - \theta_w) d\phi dr$$

where θ_w is the orientation of an arbitrary unit vector w. Then, according to the first-order optimality condition with respect to θ_w, we have

$$\int_r r^3 \int_0^{2\pi} \sin 2(\phi - \theta_w) \psi(v_{r,\phi}) dr d\phi \Bigg|_{\theta_w = \beta_i} = 0. \quad (8)$$

As shown in Fig. 3(b), both the above integrations (7) and (8) consist of two parts: one within the inscribed disk of $\mathcal{V}_i$ (the green area) and another outside (e.g., along the red strips whose angles are in $[\gamma_1, \gamma_2] \cup [\gamma_1 + \pi, \gamma_2 + \pi]$). Both the integrations upon the former part are always zero, regardless of the variables θ_i and θ_w.

For the second part, the inner integrations are carried out only within $[\gamma_1, \gamma_2] \cup [\gamma_1 + \pi, \gamma_2 + \pi]$. Given that $f(\cdot)$ preserves the rotation symmetry of order 2, our careful derivation shows that this part is zero for (7) and (8) if $\theta_i = \theta_w = (\gamma_1 + \gamma_2)/2$, hence we have $\theta_i^* = \beta_i$. □

In Section 5, we will apply these optimality conditions to design a localized iterative algorithm to obtain local maximum of CMMSDP.

4.3. Voronoi cells under new metric

Under the new directional metric $\|\cdot\|_{P_i}^2$, the representation of Voronoi cells is more complicated than their Euclidean counterparts.

Proposition 8. *Under the metric $\|\cdot\|_{P_i}^2$ the boundaries of Voronoi cells $\{\mathcal{V}_i\}$ consist of piecewise hyperbolas.*

Proof. Given two node locations, u_i and u_j, with orientation angles θ_i and θ_j respectively. Let $M_i = P_i^T P_i$, the bisector between u_i and u_j is given by the following quadratic equation

$$g(x) = (x - u_i)^T M_i(x - u_i) - (x - u_j)^T M_j(x - u_j) = 0.$$

It is well known that, under the Cartesian coordinate system, the curve of a quadratic equation in 2D is always a *conic section*. Furthermore, a conic section can be classified as ellipse, parabola, or hyperbola, according to the sign of the determinant $\triangle$ $(<, =, >,$ respectively). A careful derivation suggests $\Delta = 4(a - b)^2 \sin^2(\theta_i - \theta_j) \geq 0$ for our case. Therefore, the bisector of any two nodes is a hyperbola. As there are two branches of a hyperbola, the branch that serves as the bisector is the one that "separates" u_i and u_j, i.e., $(x - u_i)P_i^T P_j(x - u_j) < 0$, for any x on the bisector. □

However, it is interesting to note that, by unifying the orientation, the computations for constructing Voronoi cells can be highly simplified.

Proposition 9. *If all sensors have the same orientation, Voronoi cells $\{\mathcal{V}_i\}$ generated according to $\|\cdot\|_{P_i}^2$ are all convex polytopes.*

Proof. It follows from the above proof that, when $\theta_i = \theta_j$, the equation becomes $x - u_i = x - u_j$ in a linearly transformed space by P_i. So the bisector is a line and the Voronoi cells are all convex polytopes. □

We plot the Voronoi cells for both arbitrarily and uniformly oriented sensors in Fig. 4. While arbitrary orientations may result in discontinuous distorted Voronoi cells: the hatching areas in Fig. 4(a), unifying the orientation leads to continuous polygonal cells in Fig. 4(b). Later in Section 6, we use this property to devise a light-weight solution to CMMSDP.

5. Concurrent rotation and motion control

In this section, we derive a control law to concurrently drive sensor rotation and motion. This control process is performed in an iterative manner by each node, relying only on locally available information (i.e., the information acquired from its neighbors). We first present the control process in Algorithm 1 , then explain it in detail.

5.1. Building local coordinate systems

According to assumption A4, a node n_i can obtain the distances among nodes in $\mathcal{N}(n_i)$ through message exchanges, based on which, a certain 2D embedding technique (e.g., [29]) is applied to construct a local coordinate system (line 2). This step is not necessary if other positioning devices (e.g. GPS) are equipped.
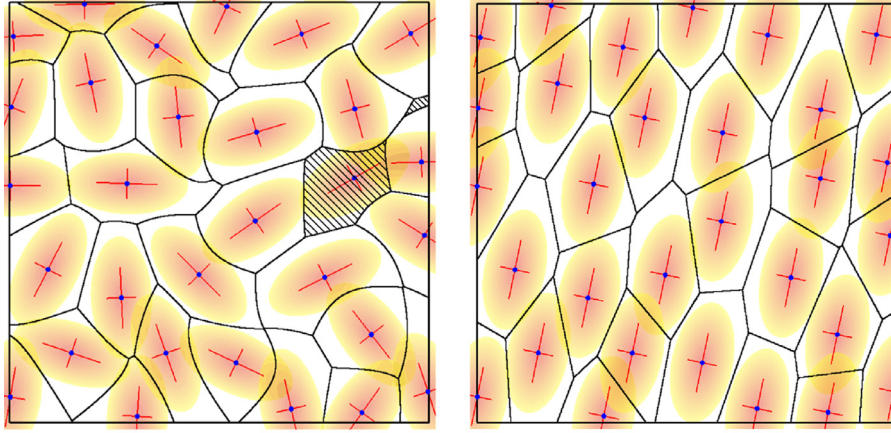
(a) Arbitrarily oriented sensors   (b) Uniformly oriented sensors

Fig. 4. Illustrating the directional Voronoi cells. The ellipses are only used to indicate directionality, rather than demarcating the sensing range.

5.2. Voronoi partition for directional sensors

In each iteration, every sensor node n_i broadcasts its current orientation θ_i to its one-hop neighbors $\mathcal{N}(n_i)$ (line 3). Being aware of location and orientation information, each node n_i may calculate the bisectors between itself and it neighbors. According to Proposition 8, the bisectors between directional sensors are hyperbolas. Therefore, what a node calculates are the parametric forms of the bisectors and their intersections. This information is sufficient to characterize the Voronoi partition $\{\mathcal{V}_i\}$ (line 4). This step is motivated by the optimality condition stated in Proposition 4.

Similar with [4], when certain bisectors are hidden from a node n_i because another node that determines this bisector is beyond the communication range of n_i, the node n_i closes the open sides of its Voronoi cell by using the transmission range as a boundary. On the other hand, as we will shown in Section 5.3, we carefully control the step sizes for the nodes to move towards the centroids; hence this problem does not constitute a threat to the algorithm convergence.

Another problem is the possible discontinuity in cells, as shown in Fig. 4(a). In general, the *main component* of $\mathcal{V}_i$ (i.e., the one that contains u_i) is much larger than other components (should they exist). Moreover, when the function $f(\cdot)$ decreases sufficiently over a large distance, approximating a Voronoi cell by its main component numerically introduces no error. As we will show in Section 7, our algorithm finally outputs a regular deployment where there is no disconnected Voronoi cell.

5.3. Motion control towards centroids

The motion control (lines 4 and 5) follows directly from the optimality condition stated in Proposition 5. The major difference between our control process and the *Centroidal Voronoi Tessellations* (CVT) [13] for omnidirectional sensors is the following. For motion control based on CVT (which mostly assume $f(x) = -x$), the centroid of $\mathcal{V}_i$ is indeed the local maximum for each iteration. Therefore, the control law proposed in [13] can be considered as a *steepest descent approach*. However, Proposition 5 is valid only for regular geometric shapes. Therefore, although we may use this proposition to show the optimality at the algorithm termination (where Voronoi cells become regular), moving towards the centroids may not be a gradient direction (though it is likely to be close to the gradient) before termination. As a result, we need to test whether a motion does increase the coverage (objective function) before an actual move (line 5).

In a practical implementation, we use the *Armijo rule* to fine tune the step size α [30]. Basically, if the objective does not increase with the current step size α, we backtrack the step size by $\alpha^+ = \eta\alpha$ for $0 < \eta < 1$. If the current sensor location is not a local maximum, there exists a point in its neighborhood that gives a higher value of the objective. Therefore, the Armijo rule always leads to a location change, unless the current location is already a local maximum. Such a strategy of tuning step size in each iteration not only guarantees the optimality of our algorithms, but also prevents sensor nodes from zigzag moving and hence serves the purpose of energy efficiency.

5.4. Rotation control based on PCA

Although the optimality condition given by Proposition 6 is sufficiently general, it is too costly to solve the integral equation (5). Therefore, we actually approximate θ_i^* by the results given in Proposition 7. The computation of the main axis (6) is essentially a continuous version of *Principal Component Analysis* (PCA) [31]. Therefore, we use PCA to numerically compute the main axis of an arbitrarily shaped Voronoi cell. Similar to the motion control, the rotation direction also may not be the gradient before termination. Therefore, we also need to test the variance in coverage before rotating (line 6). The Armijo rule, similar to the motion control, is again used to adapt the step size. Note that, each sensor node only computes and broadcasts the orientation value to its one-hop neighbors in every iteration, instead of mechanically rotating. Node

---

Algorithm 1: CRMC

Input: For each $n_i \in \mathcal{N}$, initial position u_i^0 and orientation θ_i^0, step size α, stopping tolerance ϵ

Output: $\{u_i^*\}$ and $\{\theta_i^*\}$

1 For every node $n_i \in \mathcal{N}$ periodically (every τ ms):
2   Construct a local coordinate system using the mutual distances among $\mathcal{N}(n_i) \cup \{n_i\}$
3   Broadcast θ_i to $\mathcal{N}(n_i)$
4   Compute the Voronoi cell $\mathcal{V}_i$, centroid $C_{\mathcal{V}_i}$, and main axis orientation β_i
5   **if** $|u_i - C_{\mathcal{V}_i}| > \epsilon$, $\mathfrak{C}_{\mathcal{V}_i, \theta_i}(u_i) > \mathfrak{C}_{\mathcal{V}_i, \theta_i}[u_i + \alpha(C_{\mathcal{V}_i} - u_i)]$ **then**
      $u_i^+ \leftarrow u_i + \alpha(C_{\mathcal{V}_i} - u_i)$;
6   **if** $|\theta_i - \beta_i| > \epsilon$, $\mathfrak{C}_{\mathcal{V}_i, u_i^+}(\theta_i) > \mathfrak{C}_{\mathcal{V}_i, u_i^+}[\theta_i + \alpha(\beta_i - \theta_i)]$ **then**
      $\theta_i^+ \leftarrow \theta_i + \alpha(\beta_i - \theta_i)$;

n_i adjusts its orientation to θ_i^* and performs sensing task after our algorithm achieves convergence at the end.

5.5. Performance and complexity analysis

We hereby sketch the important aspects of the algorithm performance. The convergence of CRMC is guaranteed by the ascending feature of the iteration and boundedness of the coverage function $\mathfrak{C}$. As CRMC is a distributed version of a gradient-based approach, it converges linearly (or geometrically) [30], in another word, $\lim\sup_{k\to\infty} |\mathfrak{C}_{k+1} - \mathfrak{C}^*|/|\mathfrak{C}_k - \mathfrak{C}^*| < 1$. The (local) optimality at the algorithm termination is achieved if the Voronoi cells satisfy the symmetry property stated in Section 4. As we will show in Section 7, the symmetry property is almost satisfied. Within each iteration, the computational complexity is mostly incurred by computing the centroids and main axes through numerical integrations. Both the complexity and accuracy grow linearly in the number of finite elements used to approximate a space.

6. Staged rotation and motion control

Although manipulating Voronoi cells with curved boundaries requires only basic arithmetic operations, we further simplify the computation and propose SRMC as a more efficient alternative. The basic idea is to leverage on the convex polytope cells by unifying sensor orientations. As unifying orientations can be achieved by a simple diffusion process, and Voronoi cells with linear boundaries are easier to handle, we conclude that the algorithm incurs less computational load than CRMC.

6.1. Unifying orientations

Given sensors with arbitrary orientations, what we need is a consensus protocol to reconcile the difference. Based on the results reported in [32], we devise a gossip-based protocol entailing only localized message exchanges (Algorithm 2); it reaches a consensus on the average value of all the initial orientations. Let $\Theta = [\theta_1, \dots, \theta_N]^T$ and $W = [w_{ij}]$, we know that the gossip-based unification process $\Theta^+ = W\Theta$ converges iff **1** (i.e., the all-one vector) is the eigenvector (both left and right) of W corresponding to the *spectral radius* 1 [32]. As our algorithm needs to be localized, we take $w_{ij} = \max^{-1}\{|\mathcal{N}(n_i)|, |\mathcal{N}(n_j)|\}$ and determine w_{ii} according to $W\mathbf{1} = \mathbf{1}$. Similar with CRMC, the rotation is performed mechanically only when the algorithm terminates.

6.2. Voronoi partition and motion control

The partition and control procedure here are similar to those described in Sections 5.2 and 5.3. However, due to the unified orientations, computations incurred by the control process are greatly

---

Algorithm 2: Gossip-based orientation unification.

Input: For each $n_i \in \mathcal{N}$, initial orientation θ_i^0, stopping tolerance ϵ
Output: Uniform orientation $\bar{\theta} = \sum_i \theta_i^0/N$
1 For every node $n_i \in \mathcal{N}$ periodically (every τ ms):
2 Broadcast;
3 (θ_i) to all $n_j \in \mathcal{N}(n_i)$
4 **upon** Receive;
5 ($\{\theta_i\}$) from all $n_j \in \mathcal{N}(n_i)$ **do**:
6 **if** $|w_{ii}\theta_i + \sum_{n_j \in \mathcal{N}(n_i)} w_{ij}\theta_j - \theta_i| > \epsilon$ **then**
7 $\quad \mid \quad \theta_i^+ \leftarrow w_{ii}\theta_i + \sum_{n_j \in \mathcal{N}(n_i)} w_{ij}\theta_j$
8 **end**

---

simplified. For example, the bisectors are all lines: each passes the midpoint of two neighboring nodes and is perpendicular to $f'(\cdot)$ at that point. Also, as all cells are convex polytopes, they are connected, hence the main component is equal to the cell. Additionally, it is much easier to calculate the centroids of the convex polytopes. The control law only deals with motion: it again moves a sensor towards the centroid of its cell, while adapting the step size to keep increasing the coverage objective. Although the control law does not act on the orientations anymore, the motion control and Voronoi partition are actually adapting the shapes of the cells (hence their main axes) towards the unified orientation.

6.3. Performance and complexity analysis

The linear convergence of SRMC is guaranteed by the same reason as that of CRMC. Although we cannot claim optimality for SRMC at the algorithm termination (as it sacrifices one degree of freedom in optimizing the coverage), its performance may not be worse than CRMC: remember CRMC only reaches local maximums. As we will demonstrate in Section 7, the Voronoi cells tend to adapt their main axes to the unified sensing orientation. Therefore, it is highly possible that SRMC also leads to some local maximum. Thanks to the convex polytope cells induced by a uniform orientation, the computation load for each iteration is almost negligible.

7. Evaluation

In this section, we perform extensive experiments to verify the efficacy of our algorithms in TOSSIM [33]. We first briefly introduce the simulation settings. We then study the convergence of our algorithms. Moreover, we evaluate our algorithms in terms of coverage and energy consumption, and finally illustrate the adaptivity of our algorithms to arbitrarily shaped targeted regions with diverse density.

7.1. Simulation settings

As our algorithms can be applied to any non-increasing SDF, we adopt a Gaussian-like function $f(x) = \exp(-cx/2)$ as the SDF for all simulations, where c represents the attenuation of the sensitivity. We first set the EDF $\psi(v) = 1$ in Sections 7.2 and 7.3, then we use other EDFs in Section 7.4 to verify the adaptability of our algorithms. The two metrics that we adopt to evaluate the algorithm performances are (i) *coverage*, which is the value of the objective function $\mathfrak{C}(\{\mathcal{A}_i\}, \{u_i\}, \{\theta_i\})$ and (ii) *energy consumption*, which is in proportion to moving distance. Recall that the sensor nodes adjust their orientations after the algorithm terminates; we thus omit the cost induced by rotation as it only leads to $O(1)$ energy consumption and is much less than the energy cost in moving. For N nodes to cover an area of size $|\mathcal{A}|$, considering our algorithms usually lead to regular deployments, we empirically set a limited transmission range $r = 2.5\sqrt{|\mathcal{A}|/(\pi N)}$, such that each sensor node has approximately 6 to 7 one-hop neighbors to ensure the network connectivity. In another word, this assumption can be in turn used to determine the number of nodes with a certain communication range as a result of our algorithms. We also set the sensing directionality matrix P_i (see Equation (1)) such that the induced ellipse $\mathcal{E}_i$ (centered at u_i and taking $1/a$ and $1/b$ as the major and minor radii, respectively) has an area of $|\mathcal{A}|/N$. In order to give sufficient time to conduct a movement, we set the communication round τ as 10 second.

Although we are the first to deal with directional sensors, existing algorithms for omnidirectional sensor, enhanced by our orientation handling mechanism, may also be applicable. In the following, we will compare our algorithms with SRMC-Minimax: a variance of our SRMC for which the motion direction is determined by

(a) Initial deployment

(b) Outcome of CRMC
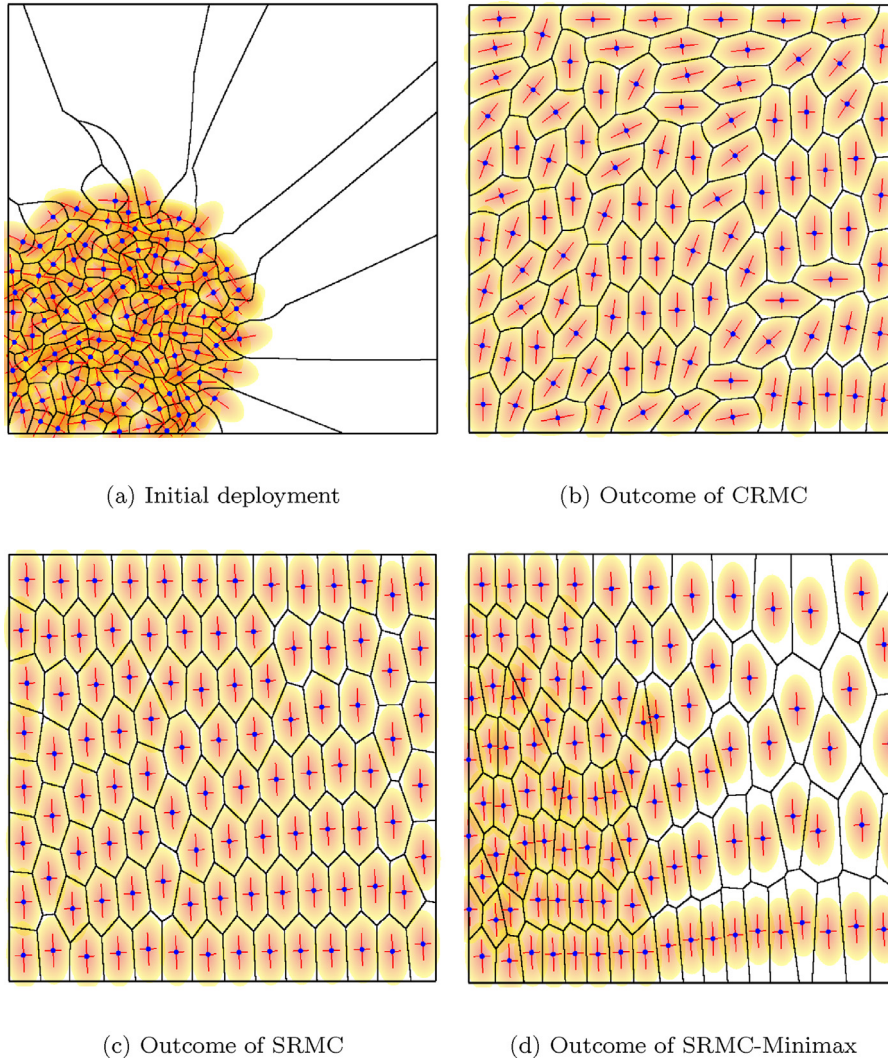
(c) Outcome of SRMC

(d) Outcome of SRMC-Minimax

Fig. 5. Initial deployment and the outcomes of different algorithms.

the minimax point (or Chebyshev center, see Section 2.1 and [4] for details) of a Voronoi cell. Though the original Minimax algorithm [4] is arguably the best control law for omnidirectional sensors, it is not directly applicable to directional sensors, because it cannot handle the rotation control (obviously) and does not converge even if the orientations are unified.

7.2. Convergence

As convergence results that we have obtained are all similar to each other, we hereby use only one such case for demonstration purpose. We consider an area of 1×1 km^2, and 100 nodes are initially deployed close to the bottom-left corner of the area, as shown by Fig. 5(a).

Then we show the outcomes of the three algorithms (assuming $c = 1$) in the remaining sub-figures of Fig. 5. It is obvious that both CRMC and SRMC lead to evenly distributed and mostly symmetric Voronoi cells, confirming the optimality results we stated earlier. On the contrary, SRMC-Minimax results in rather uneven and asymmetric cells, since the Chebyshev centers used in the context of omnidirectional sensing model [4] do not have optimality for our CMMSDP.

We also show the converging processes (coverage vs. communication rounds) of the three algorithms in Fig. 6, assuming three different values of c. First, we use the black dotted line to repre-
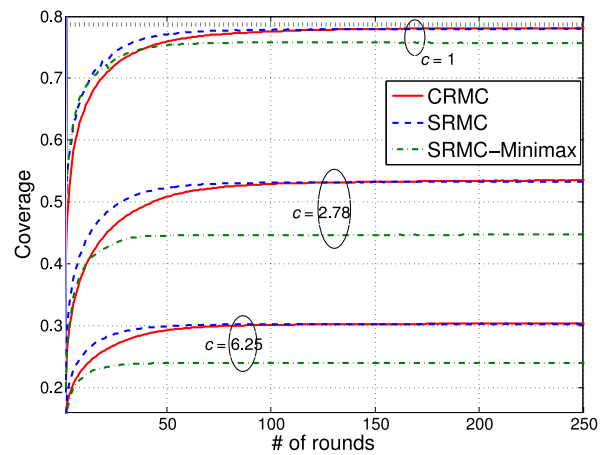

Fig. 6. Convergence of the three algorithms.

sent the naive upper bound of global optimal coverage for $c = 1$,[2] and we can observe that both CRMC and SRMC go very close to

---

[2] The bound is computed by considering every node covers the same area $|\mathcal{A}|/N$ and by maximizing individual cell coverage over all possible geometric shapes (which leads to ellipse), i.e., $\sum_{i=1}^{N} \int_{\mathcal{E}_i} e^{-\|v - u_i\|^2_{P_i}/2} dv$.
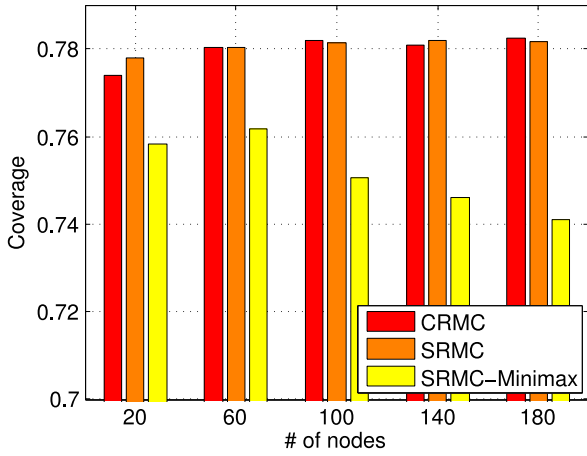
Fig. 7. Coverage of the three algorithms.
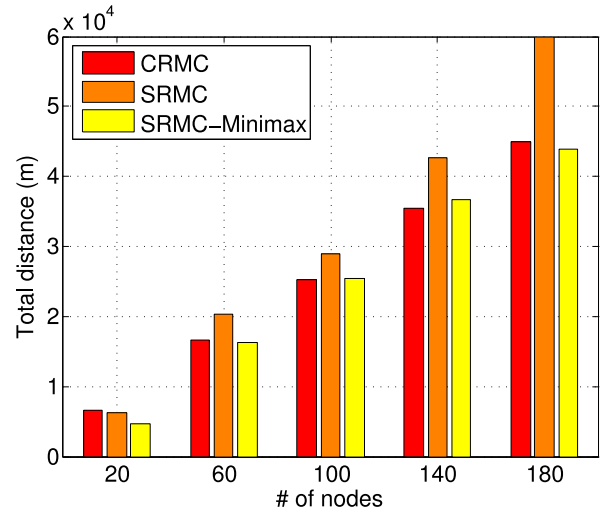
Fig. 8. Total overhead in deploying mobile nodes.

Table 1
Maximum energy consumption (Joule).

# of nodes	20	60	100	140	180
CRMC	354.6	341.1	328.1	399.7	485.3
SRMC	267.9	616.5	543.1	752.5	982.7
SRMC-Minimax	207.3	464.5	445.1	746.4	491.2

the bound, which confirms the optimality of our algorithms. Second, both SRMC and SRMC-Minimax converge faster than CRMC during the first 40 rounds. This is due to the simplification introduced by unifying the orientation before the motion control. In the long run, SRMC-Minimax performs worse than the other two, while CRMC is marginally better than SRMC. There is no surprise as the centroidal direction used by both CRMC and SRMC stems from an optimality condition (and CRMC further control rotations based on another optimality condition), whereas SRMC-Minimax is purely heuristic. The superiority of CRMC and SRMC is more conspicuous with larger c, because the impact of deployment strategy on coverage becomes more significant for sensors with more limited sensitivity.

It can be observed that both CRMC and SRMC converge in about 120 rounds. Considering CC2420 radio has transmit power 52.5 mW and receiving power 56.4 mW [34] and each transmitting/receiving operation takes around 20 ms, each sensor node spends only about 0.3 Joule on communication by running our algorithms. Moreover, although SRMC-Minimax reaches convergence in around 70 rounds (faster than CRMC and SRMC), the rounds need to interpreted differently from those in [4]. As the Minimax algorithm used in [4] (to deal with omnidirectional sensors) directly moves nodes to the minimax points of their current cells, while our algorithms adopt Armijo rule to tune the step size, the convergence of the Minimax algorithm is faster in terms of rounds. However, such a full step size leads to oscillation in mobility control for directional sensors, so we adopt a fractional step size that adapts to the geometry of current cells through backtracking (see Section 5.3). Consequently, more rounds in our case is not meant to much higher energy consumption (which we will illustrate in Section 7.3).

7.3. Coverage and energy consumption

In this section, we fix the area as 1×1 km^2 and the SDF as $f(x) = \exp(-x/2)$, but we vary the network size from 20 to 180. We show the coverage of the three algorithms in Fig. 7. As we scale the sensing directionality matrix P_i according to $|\mathcal{A}|/N$, increasing network size should slightly increase the optimal coverage (analogous to the ellipse packing with a decreasing size of each ellipse). This is clearly shown by both CRMC and SRMC in Fig. 7. Moreover, the abnormal decrease in coverage for SRMC-Minimax again confirms the non-optimality of this heuristic. Finally, as CRMC and SRMC may both converge to local maximums, one cannot be constantly superior to another.

We also evaluate the overall energy consumption by demonstrating the total moving distance until the algorithm terminations for these three algorithms. The results are shown in Fig. 8. It is obvious that, whereas CRMC and SRMC-Minimax have comparable total moving distance, SRMC moves longer than the other two algorithms. Recall that CRMC has to deal with curved Voronoi cells, the outperformance in moving distance can compensate the expense in computations to some extent. Furthermore, considering the significant advantage of SRMC over SRMC-Minimax in coverage, there is no surprising as a cost has to be paid to earn the optimality.

Additionally, we evaluate the maximum energy consumption on moving the sensor nodes in Table 1, which is driven by realistic power consumption data. We hereby assume that a mobile sensor node is equipped with a Micromo coreless DC motors (http://www.micromo.com/products/dc-motors/coreless-dc-motors-data-sheets). The power consumption of this motor is 120 mW and it may move a MicaZ Mote in a speed of 0.2 m/s. Considering a 2450 mAh Energizer (www.energizer.com) AA battery contains 33 kJ, the maximum individual node consumption only accounts for a small fraction of the node's power storage, as shown in Table 1.

7.4. Adapting to obstacles and variable density

In this section, we verify the performance of the three algorithms in adapting to a density function ψ, as well as to obstacles in a network region. Among many experiments we have performed, we choose to exhibit two scenarios by Figs. 9 and 10, respectively. We hereby use a color spectrum on the network region to represent $\psi \in [0, 1]$, with blue and red representing the lowest and highest importance, respectively. The "holes" within a network region are obstacles that nodes cannot move upon. In order to make the numbers comparable to each other, we normalize the two targeted regions to 1 km^2 areas respectively. The figures
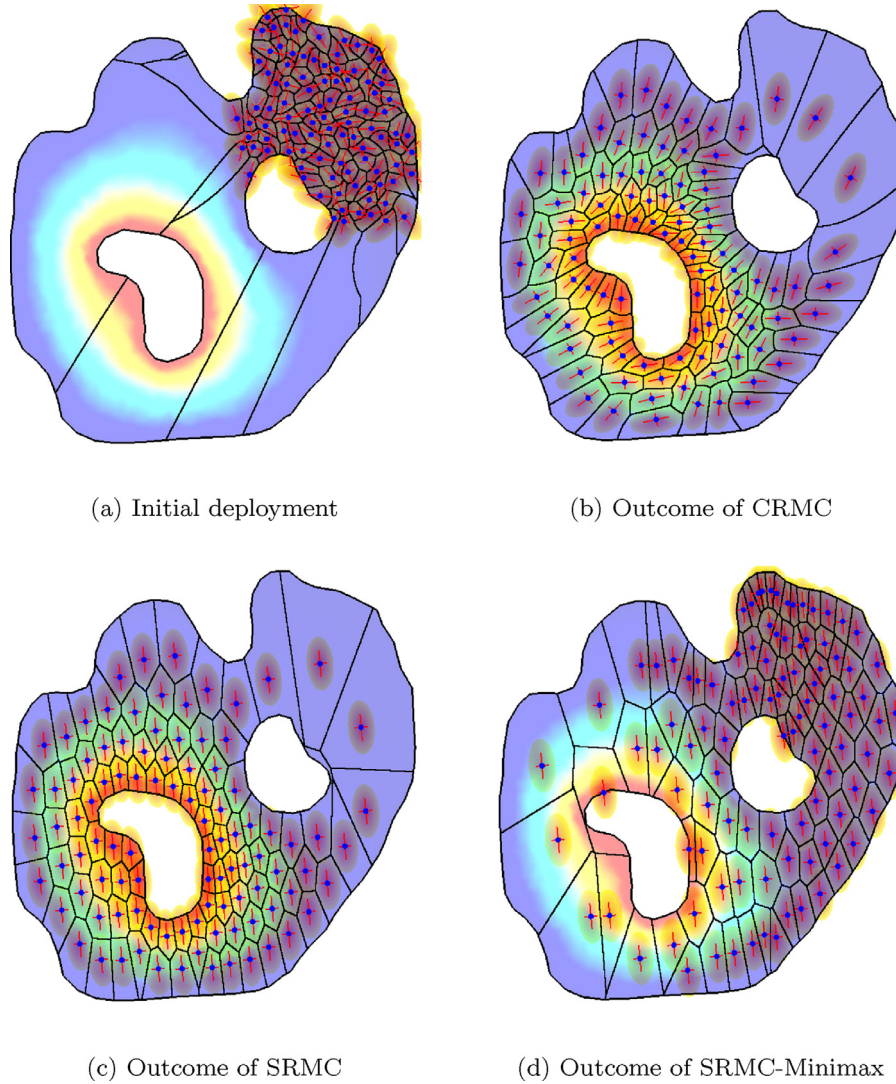
(a) Initial deployment

(b) Outcome of CRMC

(c) Outcome of SRMC

(d) Outcome of SRMC-Minimax

Fig. 9. Adaptability to obstacles and variable densities I.

Table 2
Coverage functions for two targeted areas with obstacles and variable densities.

	CRMC	SRMC	SRMC-Minimax
Deployment I	9.58e-2	9.42e-2	6.05e-2
Deployment II	9.88e-2	1.04e-1	7.03e-2

clearly show that our CRMC and SRMC both adapt well to variable densities and obstacles: they achieve almost the same coverage and more sensor nodes are located in the area with high density value. However, the performance of SRMC-Minimax is far from satisfactory.

Moreover, we use the values of coverage functions to evaluate the three deployment strategies in Table 2. Table 2 confirms the visual results from Figs. 9 and 10. Similar to the results in Fig. 7, there is no significant gap between CRMC and SRMC in terms of coverage quality, as they both converge to local minima. Meanwhile, it is obvious that CRMC and SRMC result in much better coverage than SRMC-Minimax.

Our results confirm the statement made in [7] about the inability of previous Voronoi-based approaches (e.g., [4]) to obstacles in

the sensing field. It is also demonstrated that a good coverage can be achieved with movements guided by only local information, instead of using a global geometrical structure to guide the deployment (as did in [7]).

8. Conclusion

In this paper, we address a challenging problem on maximizing the sensing coverage using mobile sensor. The contribution we make in this paper is first-of-its-kind in modelling the sensing capability as a directional distribution function which leads to proposing two optimal and practically realisable autonomous deployment strategies, CRMC and SRMC to solve the optimization problem in a localized fashion. Our theoretical analysis and experiment results have shown that, both CRMC and SRMC achieve (local) optimal coverage and adapt well to obstacles and variable densities in the region under surveillance with affordable energy and time cost.

It may be noted that, in this paper, connectivity is guaranteed by a proper density for the initial deployment, such that our algorithms lead to connected WSN deployments. One immediate extension to this paper would be on investigating the joint coverage and connectivity problem under low density.
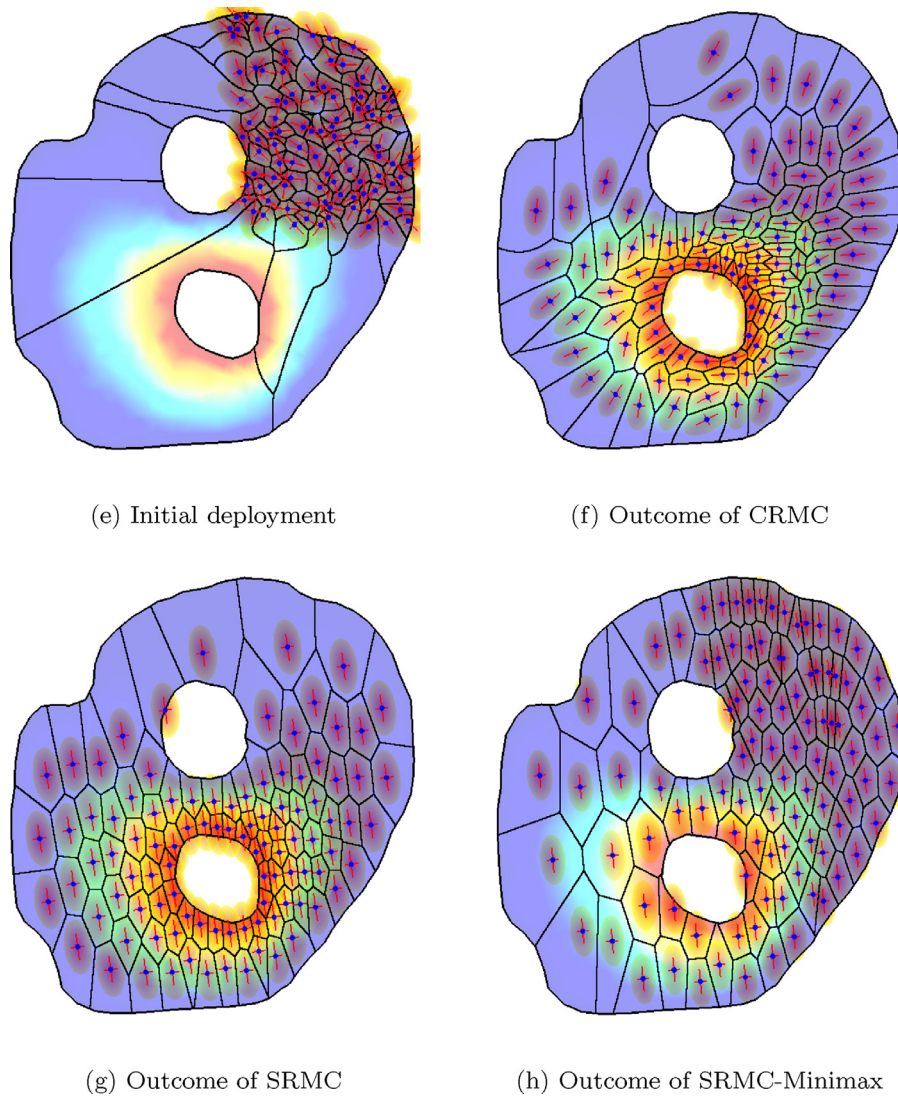
(e) Initial deployment

(f) Outcome of CRMC

(g) Outcome of SRMC

(h) Outcome of SRMC-Minimax

Fig. 10. Adaptability to obstacles and variable densities II.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.

[2] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, G. Sukhatme, Robomote: enabling mobility in sensor networks, in: Proceedings of the 4th ACM/IEEE IPSN, 2005, pp. 404–409.

[3] J. McLurkin, E. Demaine, A distributed boundary detection algorithm for multi-robot systems, in: Proceedings of IEEE/RSJ IROS, 2009. 4791–4789

[4] G. Wang, G. Cao, T.L. Porta, Movement-assisted sensor deployment, IEEE Trans. Mob. Comput. 5 (6) (2006) 640–652.

[5] N. Bartolini, T. Calamoneri, T.L. Porta, S. Silvestri, Autonomous deployment of heterogeneous mobile sensors, IEEE Trans. Mob. Comput. 10 (6) (2011) 753–766.

[6] F. Li, J. Luo, W. Wang, Y. He, Autonomous deployment for load balancing k- surface coverage in sensor networks, IEEE Trans. Wireless Commun. 14 (1) (2015) 279–293.

[7] G. Tan, S. Jarvis, A.-M. Kermarrec, Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks, IEEE Trans. Mob. Comput. 8 (6) (2009) 836–848.

[8] F. Li, J. Luo, S. Xin, W. Wang, Y. He, LAACAD: load balancing k-area coverage through autonomous deployment in wireless sensor networks, in: Proceedings of the 32nd IEEE ICDCS, 2012, pp. 566–575.

[9] M. Hawkes, A. Nehorai, Acoustic vector-sensor beamforming and capon direction estimation, IEEE Trans. Sig. Process. 46 (9) (1998) 2291–2304.

[10] S. Chowdhury, M. Ahmadi, W. Miller, Design of a MEMS acoustical beamforming sensor microarray, IEEE Sensor J. 2 (6) (2002) 617–627.

[11] M. Jin, G. Rong, H. Wu, L. Shuai, X. Guo, Optimal surface deployment problem in wireless sensor networks, in: Proceedings of the 31st IEEE INFOCOM, 2012, pp. 2345–2353.

[12] Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: applications and algorithm, SIAM Rev. 41 (4) (1999) 637–676.

[13] J. Cortés, S. Martínez, T. Karataş, F. Bullo, Coverage control for mobile Sensing networks, IEEE Trans. Robot. Autom. 20 (2) (2004) 243–255.

[14] L. Wu, H. Du, W. Wu, D. Li, J. Lv, W. Lee, Approximation for minimum connected sensor cover, in: Proceedings of the 32nd IEEE INFOCOM, 2013, pp. 1187–1194.

[15] Z. Yu, J. Teng, X. Li, D. Xuan, On wireless network coverage in bounded areas, in: Proceedings of the 32nd IEEE INFOCOM, 2013, pp. 1195–1203.

[16] H. Wang, H. Roman, L. Yuan, Y. Huang, R. Wang, Connectivity, coverage and power consumption in large-scale wireless sensor networks, Comput. Netw. 75 (24) (2014) 212–225.

[17] X. Liu, K. Wu, Y. Zhu, L. Kong, M. Wu, Mobility increases the surface coverage of distributed sensor networks, Comput. Netw. 57 (11) (2013) 2348–2363.

[18] S. He, J. Chen, X. Li, X. Shen, Y. Sun, Cost-effective barrier coverage by mobile sensor networks, in: Proceedings of the 31st IEEE INFOCOM, 2012, pp. 819–827.

[19] B. Liu, O. Dousse, P. Nain, D. Towsley, Dynamic coverage of mobile sensor networks, IEEE Trans. Parallel Distrib. Syst. 24 (2) (2013) 301–311.

[20] A. Howard, M. Mataric, G. Sukhatme, Mobile sensor networks deployment using potential fields: a distributed, scalable solution to the area coverage problem, in: H. Asama, T. Arai, T. Fukuda, T. Hasegawa (Eds.), Distributed Autonomous Robotic System 5, Springer, 2002.

[21] Y. Song, B. Wang, Z. Shi, K. Pattipati, S. Gupta, Distributed algorithms for energy-efficient even self-deployment in mobile sensor networks, IEEE Trans. Mob. Comput. 13 (5) (2014) 1035–1047.

[22] H. Imai, M. Iri, K. Murota, Voronoi diagram in the Laguerre geometry and its applications, SIAM J. Comput. 14 (1) (1985) 93–105.

[23] A. Okabe, B. Boots, K. Sugihara, S. Chiu, Spatial Tesselations: Concepts and Applications of Voronoi Diagrams, Wiley Series in Probability and Statistics, Wiley, New York, 2000.

[24] F. Aurenhammer, Power diagrams: properties, Algorithms and applications, SIAM J. Comput. 16 (1) (1987) 78–96.

[25] C. Liang, M. He, C. Tsai, Movement assisted Sensor deployment in directional sensor networks, in: Proceedings of the 6th MSN, 2010, pp. 226–230.

[26] I. Stojmenovic, D. Simplot-Ryl, A. Nayak, Toward scalable cut vertex and link detection with applications in wireless ad hoc networks, IEEE Netw. 25 (1) (2011) 44–48.

[27] S. Yoon, O. Soysal, M. Demirbas, C. Qiao, Coordinated locomotion of mobile sensor networks, in: Proceedings of the 6th IEEE SECON, 2008, pp. 126–134.

[28] P. Atkinson, Spatial statistics, in: A. Stein, F. Van der Meer, B. Gorte (Eds.), Spatial Statistics for Remote Sensing, Springer Netherlands, 2002, pp. 57–81.

[29] Y. Shang, W. Ruml, Improved MDS-based localization, in: Proceedings of the 23rd IEEE INFOCOM, 2004, pp. 2640–2651.

[30] D. Bertsekas, Nonlinear programming, 2 edition, Athena Scientific, Belmont, Massachusetts, 1999.

[31] I. Jolliffe, Principal component analysis, Springer Series in Statistics, Springer-Verlag, New York, 2002.

[32] L. Xiao, S. Boyd, Fast linear iterations for distributed averaging, in: Proceedings of the 42nd IEEE CDC, 2003, pp. 4997–5002.

[33] P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: accurate and scalable simulation of entire TinyOS applications, in: Proceedings of the 1st ACM SenSys, 2003, pp. 126–137.

[34] Chipcon's CC2420 2.4G IEEE 802.15.4/ZigBee-ready RF Transceiver, URL: http://www.ti.com/lit/ds/symlink/cc2420.pdf.

Feng Li received his PhD degree in Computer Science from Nanyang Technological University, Singapore, in 2015. Before that, he got his BS and MS degrees (both in Computer Science) from Shandong Normal University, China, in 2007, and Shandong University, China, in 2010, respectively. From 2014 to 2015, he worked as a research fellow in National University of Singapore, Singapore. He is currently an assistant professor at School of Computer Science and Technology, Shandong University, China. His research interests include computational geometry, applied optimization, distributed algorithms, wireless networking and mobile/pervasive computing.

Jun Luo got his PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland, in 2006. From 2006 to 2008, he has worked as a post-doctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School of Computer Science and Engineering, Nanyang Technological University in Singapore, where he is currently an associate professor. His research interests include wireless networking, distributed systems, multimedia protocols, network modeling and performance analysis, applied operations research, as well as network security. He is a Member of both IEEE and ACM.

Shiqing Xin got his PhD degree in Zhejiang University and now is an associate professor in Ningbo University, China. In the past several years, he focused on discrete geodesic and its applications. His research interests include computer graphics, computational geometry and mobile computing.

Ying He received the BS and MS degrees in Electrical Engineering from Tsinghua University, China, and the PhD degree in Computer Science from the State University of New York (SUNY), Stony Brook, USA. He is currently an associate professor at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the broad areas of visual computing, with a focus on the problems that require geometric computation and analysis. More information can be found at http://www.ntu.edu.sg/home/yhe.