Multihop Range-Free Localization in Anisotropic Wireless Sensor Networks: A Pattern-Driven Scheme

Qingjun Xiao, Bin Xiao, *Member*, *IEEE*, Jiannong Cao, *Senior Member*, *IEEE*, and Jianping Wang, *Member*, *IEEE*

Abstract—This paper focuses on multihop range-free localization in anisotropic wireless sensor networks. In anisotropic networks, geometric distance between a pair of sensor nodes is not always proportional to their hop count distance, which undermines the assumption of many existing range-free localization algorithms. To tolerate network anisotropy, we propose a pattern-driven localization scheme, which is inspired by the observation that in an anisotropic network the hop count field propagated from an anchor exhibits multiple patterns, under the interference of multiple anisotropic factors. Our localization scheme therefore for different patterns adopts different anchor-sensor distance estimation algorithms. The average anchor-sensor distance estimation accuracy of our scheme, as demonstrated by both theoretical analysis and extensive simulations, is improved to be better than $0.4r$ when the average sensor density is above eight, and the sensor localization accuracy thus is approximately better than $0.5r$. This localization accuracy can satisfy the needs of many location-dependent protocols and applications, including geographical routing and tracking. Compared with previous localization algorithms that declares to tolerate network anisotropy, our localization scheme excels in 1) higher accuracy stemming from its ability to tolerate multiple anisotropic factors, including the existence of obstacles, sparse and nonuniform sensor distribution, irregular radio propagation pattern, and anisotropic terrain condition, 2) localization accuracy guaranteed by theoretical analysis, rather than merely by simulations, and 3) a distributed solution with less communication overhead and enhanced robustness to different network topologies.

Index Terms—Wireless sensor networks, range-free localization, anisotropic networks.

1 Introduction

In recent years, by the advances in MEMS and communication theory, wireless sensor networks (WSNs) have revealed great potential to provide economical and practical solutions for civilian and military applications, e.g., tracking, surveillance, and environmental monitoring. In many of these applications, knowledge about sensors' geometrical positions is assumed to be an integral part of sensor readings, and it is also critical for many network protocols, e.g., topology control, clustering, and geographical routing. It thus becomes one of the fundamental issues in WSNs to acquire sensor position knowledge, called sensor localization problem.

To address this localization problem, extensive research has been conducted on multihop solutions for the following reasons. It is a naive solution to have all the sensor nodes equipped with GPS receivers to directly contact satellites, because this "one-hop" approach is prohibited by the size, cost, and power consumption constraints of sensor nodes. As a compromise, only a small portion of nodes named *anchors* have GPS receivers (or other localization equipments like laser range finder) and can know their positions accordingly, and these anchors can help to locate other "unknown" sensors. The challenging part of this anchor-based approach is that to reduce WSNs deployment cost the anchors can only be sparsely distributed. Therefore, the anchors only one hop away from a sensor may not provide enough anchor-sensor distance estimates to localize this sensor. For this reason, researchers actively seek for the multihop localization solutions that can measure anchor-sensor distances spanning multiple hops.

Among various multihop solutions, people pay great attention to the multihop range-free solutions [1], [2], [3], [4], [5], [6] that utilize only connectivity information, i.e., who is within the radio range of whom. This is because range-free solutions has no requirements for expensive ranging devices and can satisfy the accuracy requirement of many location-based applications [4]. Moreover, current ranging techniques (e.g., TDoA, AoA, and RSSI) have their inadequacies [7].

Although the previous multihop range-free solutions [1], [2], [3], [4], [5] function well in isotropic networks (that assume hop count distance between two nodes is proportional to their geometric distance), their performance deteriorates sharply in anisotropic networks. Network anisotropy stems from various factors, e.g., concave deployment region, sparse and nonuniform sensor distribution, irregular radio propagation pattern, and anisotropic deployment terrain condition. To tolerate these anisotropic factors, several methods [8], [9], [10], [11] have been proposed recently.

- *Q. Xiao, B. Xiao, and J. Cao are with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China. E-mail: {csqjxiao, csbxiao, csjcao}@comp.polyu.edu.hk.*
- *J. Wang is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, China. E-mail: jianwang@cityu.edu.hk.*

Manuscript received 28 July 2008; revised 9 Mar. 2009; accepted 17 Mar. 2010; published online 30 June 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-07-0293.
Digital Object Identifier no. 10.1109/TMC.2010.129.

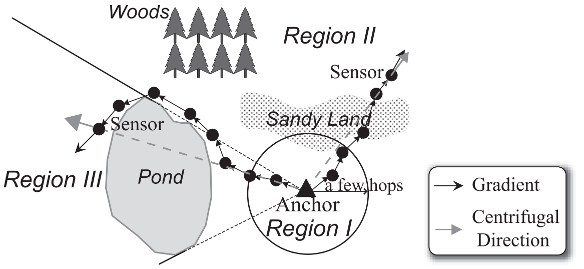

Fig. 1. Coexistence of multiple patterns in hop count field.

These methods, however, have the inadequacy to focus on only one anisotropic factor, like obstacle detour. They may also have the inadequacy of nonscalability due to the error accumulation along with the increase of network scale. Some of these methods rely on centralized computation, which consumes the microsensors' precious energies to collect and disseminate required information. Most of them neglect the impact of last hop distance on the overall distance estimation.

We focus on multihop range-free localization in anisotropic networks, and propose a distributed pattern-driven scheme to produce accurate estimates of anchor-sensor distances with the presence of multiple anisotropic factors. This accurate distance estimation is the basis of accurate sensor location estimation. The main idea of our pattern-driven scheme is to exploit the observation that the *hop count field* of an anchor (i.e., hop count distribution of sensors with respect to that anchor) can exhibit multiple patterns in an anisotropic network. One example of this coexistence of multiple patterns can be found in Fig. 1. As illustrated, region I is within a few hops from the anchor, and the hop count field there approximately exhibits a *Concentric Ring* (CR) pattern, in which sensors can approximately treat this anchor as an isotropic anchor. However, region II is far away from the anchor, and the hop count field there exhibits a *Centrifugal Gradient* (CG) pattern, in which sensors can witness the anchor as an anisotropic anchor. CR and CG patterns are different because CG pattern permits the *HopSize* (i.e., average per-hop-distance) to vary in an unpredictable manner due to the disturbance of multiple anisotropic factors, e.g., nonuniform sensor distribution, irregular radio propagation, and anisotropic terrain condition. CR and CG patterns, however, have a shared feature that a rough match is preserved between the *hop count field gradient* (with the greatest rate of increase of hop count) and the *centrifugal direction* (that departs from the anchor). The worst case is that in region III, hop count field exhibits *Distorted Gradient* (DG) pattern, in which the line-of-sight rule is violated by obstacle detour and the field gradient strongly deviates from the centrifugal direction.

To achieve accurate distance estimation in this anisotropic sensor network, we assume each sensor has the ability to classify heard anchors into three categories according to the CR, CG, and DG patterns. In practice, we put into CR category all the anchors within three (or four) hops. This threshold is chosen based on our observations of the error accumulation trend (with increased hop count) of the anchor-sensor distance estimation by isotropic algorithms (like Amorphous and DV-Hop) in rectangular anisotropic networks. It is a difficult problem to differentiate between the CG pattern (anisotropic but *slightly* detoured) and the DG pattern (anisotropic and *strongly* detoured). In fact, it maybe impossible in practice to accurately recognize the slightly detoured anchors from strongly detoured (or even from moderately detoured anchors), without the global knowledge on network topology, e.g., network boundary and obstacles shapes. A practical and efficient way is to select the eight nearest anchors into the CG category, which thus may contain detoured anchors. We choose this threshold of eight, because from the localization perspective, eight anchors are sufficient to mitigate bad anchor geometry and obtain accurate location estimate. Those anchors that are not eight nearest are put into the DG category.

The three categories of heard anchors corresponding to the three patterns (namely, CR, CG, and DG) have different dominating error sources in anchor-sensor distance estimation. For the three different categories, we therefore propose different anchor-sensor distance estimation algorithms. 1) For the CR pattern, the last hop distance is an important factor interfering the distance estimation accuracy. To reduce its impact, we propose an algorithm named CrMcs to achieve higher accuracy than DV-Hop [2] and Amorphous [3]. 2) For the CG pattern, varying $HopSize$ becomes the dominating factor and we propose the DiffTriangle to tolerate the inaccurate $HopSize$ estimates. The main idea of DiffTriangle is to revise the anchor-sensor distance estimates with the assistance from the nearest anchor to the sensor (namely *Reference Station*), which exhibits the CR pattern. Because it is inevitable for the CG category to contain detoured anchors, we enhance the DiffTriangle by $\mathrm{DiffTriangle}^*$ to tolerate obstacle detour additionally. $\mathrm{DiffTriangle}^*$ however requires two reference stations exhibiting the CR pattern. As a summary, for the CG category, when there are two reference stations in the CR category, $\mathrm{DiffTriangle}^*$ is adopted; when only one is available, DiffTriangle is used as a backup; and when there are none, CrMcs is the only choice left. 3) The surplus anchors in the DG category vulnerable to obstacle detour are dropped. Finally, when sufficient (more than six) distance estimates are collected from anchors exhibiting the CR or CG pattern, a sensor can deduce an estimate about its own location using weighted MMSE multilateration [12]

Extensive theoretical analysis on the accuracy of our pattern-driven scheme can be found in this paper. We show that, for the CR pattern, CrMcs can effectively suppress distance estimation error below $0.2r$ (r is the average radio range of sensors) when network density is higher than eight. Benefiting from these accurate estimates by CrMcs, distance estimation accuracy of DiffTriangle is improved to be better than $0.4r$ when DiffTriangle is applied to the CG pattern. The average accuracy of $\mathrm{DiffTriangle}^*$, as shown by experiments, is better than $0.5r$ when handling obstacle detour. With these accurate anchor-sensor distance estimates (even when network density is as sparse as eight), the average localization accuracy approaches $0.4r$ according to Cramér-Rao lower bound [13]. This localization accuracy can satisfy the needs of many location-dependent applications, e.g., geographical routing and tracking [4].

Our pattern-driven localization scheme differs from other anisotropy tolerating methods in several fundamental aspects.

- Higher Localization Accuracy:
  - In the CR pattern, CrMcs minimizes the impact of the last hop distance on distance estimation, which is neglected by other anisotropy tolerating methods.
  - In the CG pattern, DiffTriangle can effectively tolerate the variation of *HopSize* disturbed by multiple anisotropic factors and DiffTriangle* can additionally tolerate obstacle detour. As a comparison, most of the existing works can tolerate only one anisotropic factor (obstacle detour) and ideally assume circular radio model, dense and uniform sensor distribution, and uniform terrain condition.
  - For the DG pattern, anchors in the DG category are dropped. Therefore, our scheme can easily integrate the state-of-the-art works in secured localization, which can recognize outliers and place them into the DG category to eliminate their adverse impact.
- Less Communication Overhead: The communication overhead of our method is $O(MN)$, while that of PDM is $O(M^2N)$, where M is the number of anchors and N is the number of sensors.
- Reduced Computational Complexity: The arithmetic operations needed by our scheme includes only basic trigonometric functions, bisection root finding, and MMSE multilateration. There is no need for large matric inversion as in PDM and MDS-MAP.
- Enhanced Algorithm Robustness to Different Network Topologies: Our scheme is a distributed solution that functions well in all experimental deployment regions, including rectangular, U-shaped, and O-shaped regions. Performance of centralized algorithms (like PDM) degrades dramatically in the O-shaped regions (see Section 7.5).

The rest of the paper is organized as follows: In Section 2, we propose a localization framework to give an overall impression of our pattern-driven localization scheme. Section 3 presents CrMcs for distance estimation in the CR pattern to minimize the impact of the last hop distance. Section 4 proposes DiffTriangle algorithm to tolerate the varying *HopSize* in the CG pattern. Section 5 first quantifies the impact of DG pattern on the accuracy of DiffTriangle and then enhance DiffTriangle by DiffTriangle* to further tolerate obstacle detour. Section 6 provides the algorithm pseudocode to reproduce our simulation results. Section 7 presents experimental results, comparing our algorithm with Amorphous and PDM. Finally, we analyze related work in Section 8, and conclude our paper in Section 9.

2 Proposed Localization Framework

We present a pattern-driven localization framework, which is deployed at each sensor to estimate its location. The design of this framework is inspired by the fact that from the perspective of an sensor, hop count fields of different anchors may exhibit different patterns, disturbed by different anisotropic factors. To classify heard anchors according to the three patterns (CR—isotropic, CG—anisotropic but slightly detoured, and DG—strongly detoured) and to invoke suitable distance estimation algorithms for different patterns, we present the localization framework in Fig. 2, which depicts both the data flow and control flow.

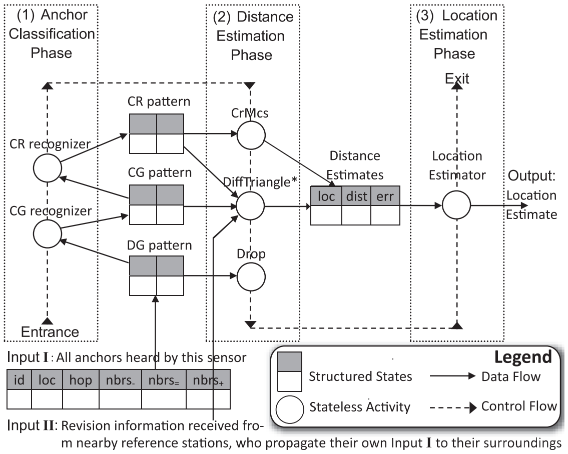
(1) Anchor Classification Phase
(2) Distance Estimation Phase
(3) Location Estimation Phase
Exit
CR pattern
CrMcs
CR recognizer
CG pattern
DiffTriangle*
Distance Estimates
loc
dist
err
Location Estimator
Output: Location Estimate
CG recognizer
DG pattern
Drop
Entrance
Input I : All anchors heard by this sensor
id
loc
hop
nbrs-
nbrs=
nbrs+
Legend
Structured States
Data Flow
Stateless Activity
Control Flow
Input II: Revision information received from nearby reference stations, who propagate their own Input I to their surroundings

Fig. 2. Localization framework deployed at each sensor.

From the perspective of control flow, our framework has only one control thread, which consists of three consecutive phases: 1) Anchor Classification Phase, 2) Distance Estimation Phase, and 3) Location Estimation Phase. In phase 1, we classify the anchors heard by a sensor into three categories (CR, CG, or DG patterns). In phase 2, we schedule a corresponding anchor-sensor distance estimator for each anchor category, which prepares a sufficient number of distance estimates for the next phase. In phase 3, we deduce an estimate about the sensor's location by weighted multilateration.

From the perspective of data flow, the framework has one output (the final location estimate) and two inputs (Inputs I and II). First, to obtain the Input I (stored temporally in the DG category), each anchor initiates networkwide flooding and each sensor hearing the flooding records the information about the anchor, such as its identifier, location, and hop count. Second, each sensor locally broadcasts its incomplete Input I (the hop counts to all heard anchors) to its immediate neighborhood, since the Input I additionally requires the knowledge on the number of neighbors having smaller (equal, and larger) hop counts than the sensor itself. Finally, each anchor propagates its Input I to three (or four) hops neighborhood by a confined flooding, which is recorded by nearby sensors as their Input II and is needed by our DiffTriangle* algorithm. Therefore, we regard our localization scheme as a distributed solution with only one simple communication protocol—the confined flooding, since the networkwide flooding and the local broadcast are both special cases of confined flooding.

In phase 1, we classify the set of all heard anchors into three subsets (namely, CR, CG, and DG) by a sequential execution of two pattern recognizers. First, the CG recognizer relocates undetoured anchors from the DG category into the CG category, since the CG pattern in contrast to the DG pattern has the slight detour assumption. This paper configures the CG recognizer to trust the nearest eight anchors, which reduces the chance to contain strongly detoured anchors in the CG category. Second, the CR

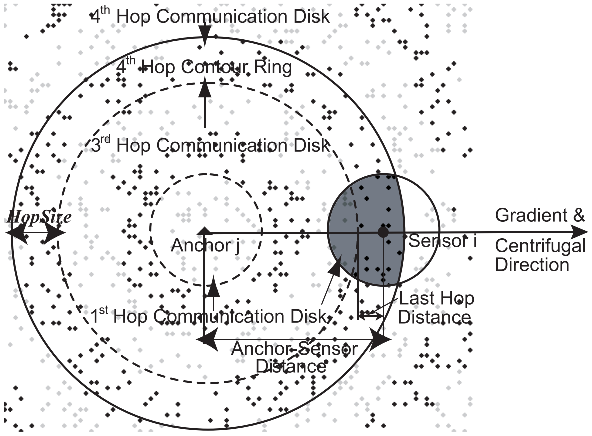

Fig. 3. Distance estimation by concentric rings pattern.

recognizer relocates anchors exhibiting the CR pattern from the CG category into the CR category, since the CR pattern additionally assumes network isotropy (with identical *HopSize*) compared with the CG pattern. For CR recognizer, we use an empirical rule to trust anchors within a few hops (i.e., three or four hops adjustably).

In phase 2, we adopt different anchor-sensor distance estimators for different anchor categories (CrMcs for the CR pattern, DiffTriangle* and DiffTriangle for the CG pattern) and drop the remaining anchors in the DG pattern. CrMcs can solve the last hop distance estimation problem for isotropic anchors and it is discussed in Section 3. DiffTriangle can tolerate varying *HopSize* and we present DiffTriangle in Section 4. DiffTriangle* is an enhancement to DiffTriangle to additionally tolerate obstacle detour and we cover it in Section 5. In phase 3, we adopt the weighted multilateration [12] as the location estimator, which is described in Section 5.2.

An advantage of our framework is its flexibility, which permits the "Stateless Activities" in Fig. 2 to be replaced by other algorithms. One example is to replace CrMcs by MDS-MAP [14], which however incurs higher communication overhead. Another example is to use the GridVoting [15] method as the CG recognizor, which can filter both obstacle detour and malicious attacks. However, secured localization is not the focus of this paper and we find, in practice, that GridVoting can only detect strongly biased anchors but not moderately detoured anchors. It maybe impossible to completely rule out detoured anchors in the CG category, and DiffTriangle* is still necessary even when the GridVoting method is employed. As a summary, our framework has the flexibility to accommodate different algorithm combinations, which can facilitate future studies in sensor network localization.

TABLE 1
Symbols and Their Descriptions

Networks	
r	average communication range of sensors and anchors
loc_i	location estimate of sensor i. If sensor i is an anchor, this knowledge is precise; otherwise, it is imprecise.
$d_j(i)$	the real geometric distance between sensor i and anchor j
$h_j(i)$	the hop count of sensor i with respect to anchor j
$N(i)$	the set of immediate neighbors of sensor i, including i
SD	Sensor Density = $\overline{\lvert N(i) \rvert}$, i.e., the expected number of immediate neighbors of a sensor
DV-Hop	
$\hat{d}_{1j}(i)$	an estimate of $d_j(i)$ given by DV-Hop
d_{hop}	average distance per hop or *HopSize*
Amorphous	
$\hat{d}_{2j}(i)$	an estimate of $d_j(i)$ given by Amorphous
$\overline{h}_j(i)$	the smoothed hop count by Amorphous
CrMcs	
$\hat{d}_{3j}(i)$	an estimate of $d_j(i)$ given by CrMcs
$r(h)$	the radius of h^{th} hop communication disks
$dsk_i(h)$	h^{th} hop communication disk of sensor i; $dsk_i(h)$ centers at sensor i, with radius $r(h)$
$a_j(i)$	the intersected area of $dsk_i(1)$ and $dsk_j(h_j(i))$
$N_j(i)$	$\{\, l \mid l \in N(i) \ \wedge \ h_j(l) \leq h_j(i) \,\}$
DiffTriangle	
$\hat{d}_{4j}(i)\vert_k$	an estimate of $d_j(i)$ revised by reference station k
$\overrightarrow{gd_j(i)}$	the gradient of anchor j's hop count field at sensor i
$\overrightarrow{s_i s_j}$	the vector from sensor i to sensor j
DiffTriangle*	
$\hat{d}_{5j}(i)\vert_{k^*}^{k}$	an estimate of $d_j(i)$ revised by reference stations k and k^*
$\gamma_j(i)$	the deviation angle between $\overrightarrow{gd_j(i)}$ and $\overrightarrow{s_j s_i}$

3 CR Pattern and Proposed CrMcs

This section presents the CrMcs algorithm to estimate the multihop anchor-sensor distance for isotropic anchors approximately exhibiting the Concentric Ring pattern. The CR pattern applies in isotropic networks and isotropic regions (like the region I in Fig. 1), in which the field gradient roughly matches centrifugal direction and *HopSize* is identical in all directions as shown in Fig. 3. We first identify the last hop distance problem—an important factor interfering the distance estimation in the CR pattern, and then show the ineffectiveness of DV-Hop and Amorphous to handle this problem. In order to minimize its impact, CrMcs is proposed to reduce distance estimation error of the CR pattern to below $0.2r$ (when network density is above eight). We give out this accuracy bound by theoretical analysis and this accuracy is crucial to guarantee satisfactory performance of DiffTriangle, which will be discussed in Section 4. We summarize the symbols in Table 1, which are used throughout the paper.

3.1 CR Pattern and the Last Hop Distance

It is well known that in networks where sensors are uniformly distributed, the hop count field of an anchor approximately demonstrates the CR pattern, under the assumption that the RF transmitters of wireless sensor nodes have a rotationally symmetric range. We argue that when estimating anchor-sensor distances using the CR pattern, another important factor influencing estimation accuracy (besides the accuracy in *HopSize* estimation) is the last hop distance.

Definition 1 (Last Hop Distance). *When $h_j(i) = 1$, the last hop distance of sensor i is $d_j(i)$, and when $h_j(i) > 1$, the last hop distance of sensor i is the shortest distance from the contour ring with hop $h_j(i) - 1$ to the sensor i.*

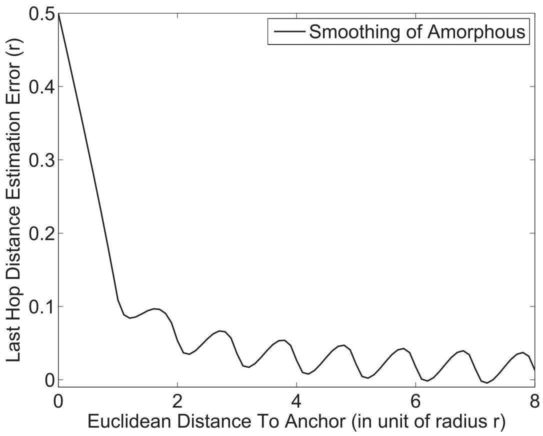

Fig. 4. Systematic error of smoothed hop in Amorphous.

If we cannot effectively estimate the last hop distance, the average distance estimation error will exceed one quarter of *HopSize* ($\approx \frac{1}{4} \cdot 0.8r = 0.2r$). This is because the maximum estimation error is half of *HopSize*, since all sensors in a contour ring (e.g., the fourth hop contour ring in Fig. 3) have the same hop count. All the sensors in this ring thus estimate the anchor-sensor distance as the mean of the corresponding ring's inner radius and outer radius.

Although this problem of last hop distance is important, it is inappropriately handled by traditional distance estimation algorithms, including DV-Hop [2] and Amorphous [3]. The DV-Hop, using the following equation, neglects this problem.

$$\hat{d}_{1j}(i) = h_j(i) \cdot d_{hop}.$$

The Amorphous, though striving to mitigate the impact of this problem using a method called "smoothing," produces biased distance estimates for the first two hops even when *HopSize* is accurately known, according to the following analysis. Amorphous adopts the smoothed hop count $\overline{h}_j(i)$, rather than the raw hop count $h_j(i)$, to derive distance estimate $\hat{d}_{2j}(i)$.

$$\hat{d}_{2j}(i) = \overline{h}_j(i) \cdot d_{hop}. \tag{1}$$

The smoothed hop count $\overline{h}_j(i)$ is calculated by a local averaging around sensor i's immediate neighborhood.

$$\overline{h}_j(i) = \frac{1}{|N(i)|} \sum_{l \in N(i)} h_j(l) - 0.5. \tag{2}$$

We analyze the systematic error of Amorphous incurred by the last hop distance in Appendix A and plot its result in Fig. 4, which indicates Amorphous can be inaccurate in the first two hops. A similar conclusion can be drawn from the simulation results in Section 7.

3.2 Proposed CrMcs

We propose the CrMcs algorithm to achieve accurate anchor-sensor distance estimation in the CR pattern. The main idea underlying CrMcs is to minimize the impact of the last hop distance by exploiting the uniform sensor distribution around a sensor's immediate neighborhood (i.e., essentially Monte Carlo sampling). Therefore, from the percentage of neighbors with no larger hop count than the sensor itself, we can estimate the area of the gray region illustrated in Fig. 3. With this area estimate, we can derive an anchor-sensor distance estimate that includes the last hop distance.

The first step of CrMcs is to estimate the area of the mentioned gray region, which is the intersected region of anchor j's $h_j(i)$ hop disk—$dsk_j(h_j(i))$—and sensor i's one hop disk $dsk_i(1)$. We represent its area by $a_j(i)$ and estimate the area by the following equation:

$$a_j(i)/\pi r^2 \approx |N_j(i)|/|N(i)|. \tag{3}$$

Its basic idea is the Monte Carlo sampling, treating each node around sensor i's neighborhood as an independent sampling. In this way, the ratio of $a_j(i)$ to the area πr^2 of sensor i's neighborhood $dsk_i(1)$ can be approximated to the proportion of sensor i's neighbors with hop counts equal to or lower than the sensor i's hop count $h_j(i)$.

The second step of CrMcs is to estimate the radius of the two intersecting disks—$dsk_j(h_j(i))$ and $dsk_i(1)$ by (4), assuming the CR pattern.

$$r(h) \approx (h-1) \cdot d_{hop} + r. \tag{4}$$

In this equation, the radius of $dsk_j(h_j(i))$ is equal to $r(h_j(i))$ and radius of $dsk_i(1)$ is $r(1)$. In (4), we intentionally assign the one hop disk's radius $r(1)$ as the sensors' average radio range r, which is determined when the sensors are deployed. This assignment improves estimation accuracy of $r(1)$, while it brings error to estimation of $r(h)$, when $h > 1$. This error is negligible in simulation, since it is quite difficult to estimate *HopSize* (or d_{hop}) precisely in practice due to the existence of radio irregularity (enlarged by the long range link) and it is always underestimated by Kleinrock's equation [16].

The final step is to estimate the anchor-sensor distance $d_j(i)$ by (5), as the distance between centers of two disks $dsk_j(h_j(i))$ and $dsk_i(1)$.

$$\hat{d}_{3j}(i) = \mathcal{A}^{-1}\ [r(1),\ r(h_j(i)),\ a_j(i)]. \tag{5}$$

This equation assumes that we have the estimates about the radius of the two disks and the area of their intersected region in the previous two steps . The function $\mathcal{A}^{-1}$ is the inverse function of $\mathcal{A}$ in (6), which is established by applying bisection root finding algorithm to function $\mathcal{A}$.

$$\begin{aligned} a_1 &= \mathcal{A}\ (r_1,\ r_2,\ d) = r_1 \cdot \frac{\theta_1 - sin\theta_1}{2} + r_2 \cdot \frac{\theta_2 - sin\theta_2}{2}, \\ \theta_1 &= 2\arccos\frac{x_1}{r_1}, \qquad x_1 = \frac{d}{2} + \frac{r_1^2 - r_2^2}{2d}, \\ \theta_2 &= 2\arccos\frac{x_2}{r_2}, \qquad x_2 = \frac{d}{2} + \frac{r_2^2 - r_1^2}{2d}. \end{aligned} \tag{6}$$

$\mathcal{A}\ (r_1,\ r_2,\ d)$ is a geometric function calculating the area of the intersected region of two disks and taking three parameters—the radius r_1, r_2 of the two intersecting disks and the distance d between their centers. The symbols used in (6) are illustrated by Fig. 5.

3.3 Error Characteristics of CrMcs

Sensor density is an important factor influencing the accuracy of CrMcs, which directly decides the accuracy of the intersected area estimation in (3). For this issue, we

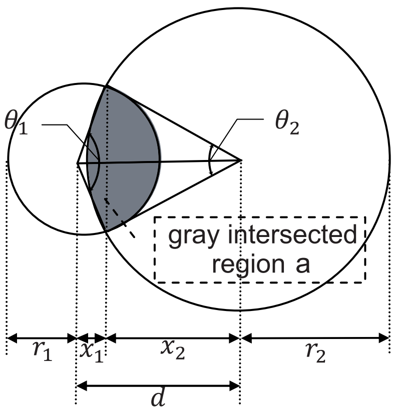

Fig. 5. The area of the intersected region of two disks.

provide an analysis in Appendix B and plot the analysis result in Fig. 6, showing that the accuracy of CrMcs is better than $0.2r$ when sensor density >8. This analysis result is consistent with the simulation result in Section 7. The limitation of this analysis, however, is its assumption of accurate *HopSize* (simplified to r) and it thus is only valuable for low hop count cases, in which the impact of inaccurate *HopSize* on the accuracy of (4) is minimized. For large hop count cases, it is inevitable for the accuracy of CrMcs (and Amorphous) to degrade with the increase of hop count (namely *error accumulation*), since the inaccuracy in *HopSize* get amplified by large hop count in (4). It is the topic of the next section on how to tolerate this error accumulation due to varied and inaccurate *HopSize*. Additionally, the accuracy of CrMcs with the presence of radio irregularity (by DOI model [4]) has been investigated by simulations in Section 7.3.

4 CG Pattern and DiffTriangle

This section considers the distance estimation problem in anisotropic networks with the presence of various anisotropic factors apart from large obstacles. In this type of networks and regions (like region II in Fig. 1), the hop count fields exhibit the *CG* pattern. The CG pattern, compared with the CR pattern,

- relaxes the assumption about identical *HopSize* to permit it to vary in all directions,

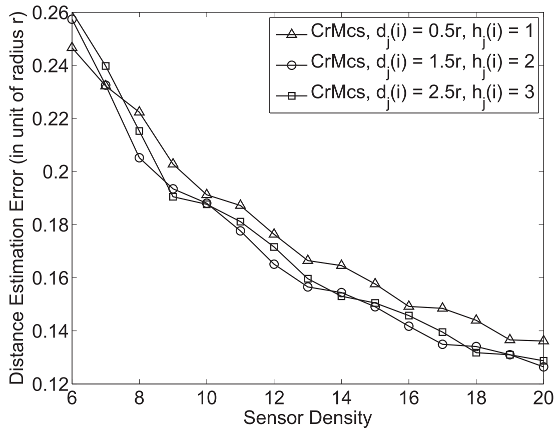

Fig. 6. Accuracy of CrMcs in different sensor densities.

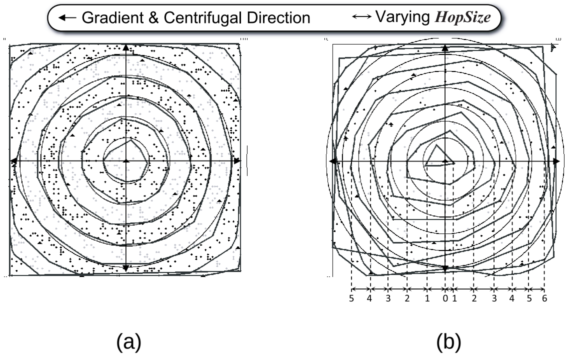

Fig. 7. Anisotropy caused by low sensor density. (a) Dense network with degree = 30. (b) Sparse network with degree = 8.

- but it preserves the assumption about the rough match between gradient and centrifugal direction (i.e., the slight obstacle detour assumption).

Therefore, isotropic algorithms assuming the CR pattern (like DV-Hop and CrMcs) encounter performance degradation in the CG pattern. To tolerate the varying *HopSize*, we propose DiffTriangle algorithm that can reduce the distance estimation error below $0.4r$ (sensor density ≥ 8). We also provide theoretical analysis for this claimed accuracy in this section.

4.1 CG Pattern with Slight Obstacle Detour

Effective localization remains a problem in sparse networks where the sensor density falls in the range of 6 to 15. There are two reasons for using this range. First, Nagpal et al. suggest in [3] that 15 is a critical minimum sensor density for Amorphous to obtain good accuracy. Second, Kleinrock and Silvester prove in [16] that 6 is the optimum sensor density to maintain the network connectivity. The localization problem in sparse networks deserves investigation, because lower sensor density implies lower deployment cost, smaller possibility of traffic jam, and radio interference.

We argue that the underperformance of Amorphous in sparse networks is caused by network anisotropy. To visualize this anisotropy, we use convex hulls to contain all sensors with the same hop count, which offer a good approximation and illustration of contour curves in a hop count field. As in Fig. 7a, when the sensor density is 30, a tight match can be seen between the concentric rings and the convex hulls, which are represented as the polygons in Fig. 7. However, when the sensor density is as low as 8, the convex hulls deviate from concentric rings, with *HopSize* varying unpredictably in different directions. This deviation explains why localization algorithms assuming network isotropy suffer from severe performance degradation in sparse networks.

A CG pattern can be extracted from the hop count fields of sparse networks. In Fig. 7b, although the *HopSize* varies when density is 8, the field gradient still roughly matches the centrifugal direction, which is drawn as the rough perpendicularity between centrifugal directions and contour curves. This type of anisotropy is summarized as the CG pattern that assumes varying *HopSize* and a rough match between field gradient and centrifugal direction.

The existence of the CG pattern in a sparse network can be explained as its nonuniform sensor distribution tendency,

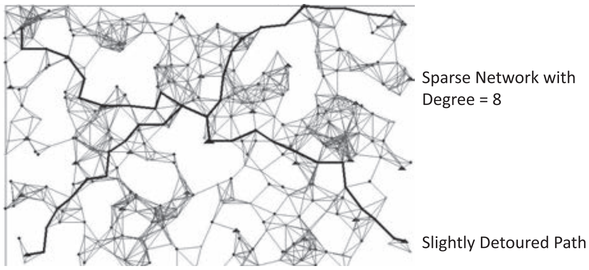

Fig. 8. Disturbance by small holes in sparse networks.

which creates numerous small holes scattered over the whole network, as illustrated in Fig. 8. These small holes distort the shortest path between the anchor and a sensor, which thus slightly deviates from the straight line connecting the two sensor nodes. Therefore, in Fig. 7b, the *HopSize* varies unpredictably in different directions, but the field gradient is only slightly disturbed from its outward direction, due to the small scale of these holes. This rough match between field gradient and centrifugal direction (the slight detour assumption) is a common trait observable in more generalized network settings, additionally assuming the presence of anisotropic terrains condition, nonuniform sensor distribution, irregular radio propagation, and inconsistent sensor radio range.

4.2 Proposed DiffTriangle

To tolerate the unpredictable variation of *HopSize* in the CG pattern and produce accurate estimates about anchor-sensor distances, we propose DiffTriangle exploiting the rough match between field gradient and centrifugal direction. The DiffTriangle gets its inspiration from the Voronoi diagram with geometrically distributed anchors acting as the sites of the Voronoi cells. Sensors within a Voronoi cell adopts the dominating anchor as their *Reference Station* to revise their distance estimates (to other distant anchors), which deteriorate due to network anisotropy if applying CrMcs or Amorphous.

We assume that the dominating reference station of a Voronoi Cell approximately exhibits the CR pattern to the sensors within the cell. This implies that these reference stations should appear in normal sensors' CR category and thus the distance from sensors to their dominating reference stations should be no more than three (or four) hops. This assumption implicitly places a demand for anchor distribution density. As an approximate estimation, to guarantee the availability of reference station in the CR category when the sensor density is 10, the anchor percentage ($=\frac{\text{Anchor Number}}{\text{Sensor Number}}$) should be roughly

$$\frac{\textit{One Anchor}}{(\text{Sensor Density}/\pi r^2)\cdot\pi(3r)^2}=\frac{1}{10}\frac{\pi r^2}{\pi(3r)^2}\approx 1.1\%.$$

Therefore, in our simulations, a random distribution of anchors with anchor percentage of 3-5 percent can guarantee the availability of reference stations for a large majority of sensors. In those "unavailable" rare cases, CrMcs is used as a backup for DiffTriangle.

The DiffTriangle algorithm gets its abbreviated name from *Differential Triangle*, since frequently the anchor, sensor, and reference station (call it *station* for short) are not collinear but are positioned as a triangle depicted by Fig. 9. In this triangle, we solve the problem of how to revise the estimate of anchor-sensor distance $d_j(i)$, benefiting from:

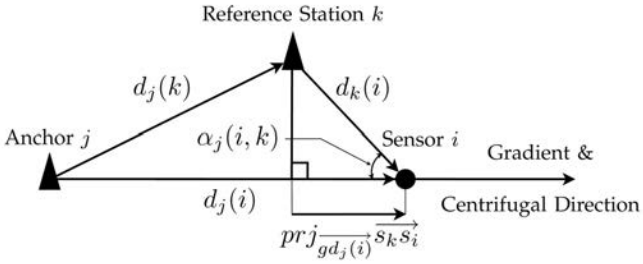

Fig. 9. DiffTriangle algorithm in the CG pattern.

1. the precisely known $d_j(k)$ from geometric coordinates of anchor j and reference station k,
2. the accurate estimate $\hat{d}_{3k}(i)$ by CrMcs of the station-sensor distance $d_k(i)$, since the nearby reference station k exhibits the CR pattern to sensor i,
3. accurately estimated

$$prj_{\overrightarrow{gd_j(i)}}\overrightarrow{s_ks_i}$$

from the proximity difference of reference station k and sensor i in anchor j's hop count field. Its estimation suffers much less from varying *HopSize* than estimation of $d_j(i)$, thanks to the geometric closeness of reference station k to sensor i.

We adopt the following equations for DiffTriangle Algorithm for CG pattern:

$$\begin{cases}\hat{d}_{4j}(i)|_k=\sqrt{d_j^2(k)-\hat{d}_{3k}^2(i)+p\hat{r}j^2_{\overrightarrow{gd_j(i)}}\overrightarrow{s_ks_i}}+p\hat{r}j_{\overrightarrow{gd_j(i)}}\overrightarrow{s_ks_i},\\ p\hat{r}j_{\overrightarrow{gd_j(i)}}\overrightarrow{s_ks_i}\ =\ [\overline{h}_j(i)-\overline{h}_j(k)]\cdot d_{hop}.\end{cases}\tag{7}$$

The $\hat{d}_{4j}(i)|_k$ in (7) is our estimate of anchor-sensor distance by DiffTriangle. The station-sensor distance estimate $\hat{d}_{3k}(i)$ required by (7) is obtained by applying CrMcs algorithm in (5). The

$$p\hat{r}j_{\overrightarrow{gd_j(i)}}\overrightarrow{s_ks_i}$$

is the proximity difference of sensor i and station k. The adopted proximity $\overline{h}_j(i)$ and $\overline{h}_j(k)$ is the smoothed hop counts of sensor i and station k (see (2)).

Our derivation of the equations of DiffTriangle is presented below. The anchor-sensor distance $d_j(i)$ can be calculated using the following trigonometric formula, assuming the angle $\alpha_j(i,k)$ is known.

$$d_j(i)=\sqrt{d_j^2(k)-d_k^2(i)\sin^2\alpha_j(i,k)}+d_k(i)\cos\alpha_j(i,k).\tag{8}$$

The angle $\alpha_j(i,k)$ can be estimated from the projection $prj_{\overrightarrow{s_js_i}}\overrightarrow{s_ks_i}$ of vector $\overrightarrow{s_ks_i}$ over vector $\overrightarrow{s_js_i}$, where $\overrightarrow{s_ks_i}$ is the vector pointing from station k to sensor i and $\overrightarrow{s_js_i}$ is the centrifugal vector from anchor j to sensor i.

$$\alpha_j(i,k) = \arccos \frac{prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i}}{d_k(i)}.$$

The critical $prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i}$ can be approximated to

$$prj_{\overrightarrow{gd_j(i)}} \overrightarrow{s_k s_i},$$

under the assumption that the centrifugal direction $\overrightarrow{s_j s_i}$ roughly matches the direction of gradient $\overrightarrow{gd_j(i)}$ of hop count field of anchor j seen by sensor i.

$$\alpha_j(i,k) \approx \arccos \frac{prj_{\overrightarrow{gd_j(i)}} \overrightarrow{s_k s_i}}{d_k(i)}. \tag{9}$$

We estimate the projection $prj_{\overrightarrow{gd_j(i)}} \overrightarrow{s_k s_i}$ in (7) by assuming

$$prj_{\overrightarrow{gd_j(i)}} \overrightarrow{s_k s_i}$$

proportional to the proximity difference $h_j(i) - h_j(k)$ of sensor i and station k, since the isotropy can be well preserved in the small region between sensor i and station k. To achieve higher accuracy, (7) uses smoothed hop count described in (2) for the proximities of sensor i and station k to solve the last hop distance problem, because smoothed hop count is accurate when hop count is larger than two (see Fig. 4).

4.3 Error Characteristics of DiffTriangle

In this section, we demonstrate by analysis that DiffTriangle can reduce the average distance estimation error to below $0.4r$ in CG pattern, when the sensor density is higher than 8. The analysis is the theoretical foundation of the weighted multilateration adopted by our localization scheme, which configures the expected error of CrMcs as $0.2r$ and that of DiffTriangle (or its enhancement DiffTriangle^*) as $0.4r$.

We linearize the impact of several factors on distance estimation error $\Delta d_j(i)$ by applying the Taylor expansion on (8): 1) station-sensor distance estimation error $\Delta d_k(i)$, and 2) angle estimation error $\Delta\alpha_j(i,k)$, which leads to the establishment of the following equation:

$$\begin{aligned}\Delta d_j(i) \approx & \left[\cos\alpha_j(i,k) - \frac{d_k(i)\ \sin^2\alpha_j(i,k)}{\sqrt{*}}\right] \cdot \Delta d_k(i) \\ & + \left[\sin\alpha_j(i,k) + \frac{d_k(i)\ \sin 2\alpha_j(i,k)}{2\sqrt{*}}\right] d_k(i) \cdot \Delta\alpha_j(i,k), \\ & \text{where } * = d_j^2(k) - d_k^2(i)\ \sin^2\alpha_j(i,k).\end{aligned}$$

To simplify our analysis, the two items in the form of $\frac{d_k(i)\dots}{\sqrt{*}}$ are approximated to 0, leading to the simplified representation of $\Delta d_j(i)$ in (10).

$$\Delta d_j(i) \approx \cos\alpha_j(i,k)\Delta d_k(i) + \sin\alpha_j(i,k) d_k(i)\Delta\alpha_j(i,k). \tag{10}$$

This approximation is reasonable because station-sensor distance $d_k(i)$ is much smaller than the anchor-station distance $d_j(k)$. The required $\Delta\alpha_j(i,k)$ in (10) can be derived by applying Taylor Expansion to (9).

$$\Delta\alpha_j(i,k) = \frac{\cos\alpha_j(i,k)\ \Delta d_k(i) - \Delta prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i}}{d_k(i)\ \sin\alpha_j(i,k)}.$$

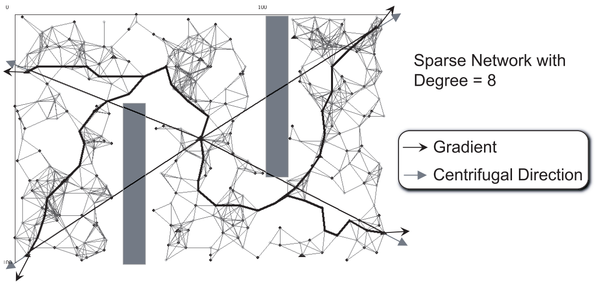

Fig. 10. Deviation of gradient and centrifugal direction.

Therefore, (10) can be converted to a more simplified form in (11).

$$\Delta d_j(i) \approx 2\ \cos\alpha_j(i,k)\ \Delta d_k(i) - \Delta prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i}. \tag{11}$$

This equation indicates that distance estimation error $\Delta d_j(i)$ of DiffTriangle is influenced by two factors: 1) estimation error $\Delta d_k(i)$ of the station-sensor distance, and 2) estimation error

$$\Delta prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i}$$

of the station-sensor proximity difference. The first factor $\Delta d_k(i) < 0.2r$, since the estimation of $d_k(i)$ is achieved using CrMcs, whose accuracy has been analyzed in the previous section. The second factor

$$\Delta prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i} < 0.2r,$$

since estimation algorithm of

$$prj_{\overrightarrow{s_j s_i}} \overrightarrow{s_k s_i}$$

by (7) minimizes the impact of last hop distance, which is similar to CrMcs. Therefore, according to (11), average distance estimation error $\Delta d_j(i)$ of DiffTriangle in convex anisotropic networks is smaller than $[1 + 2\cos\alpha_j(i,k)] \cdot 0.2r \approx 0.4r$, which is consistent with the simulation results in Section 7.

5 DG Pattern and DiffTriangle*

This section focuses on the distance estimation problem in anisotropic networks additionally assuming the presence of large obstacles. These obstacles can distort field gradients to strongly deviate from the centrifugal directions (see region III in Fig. 1, and Fig. 10). This strong deviation between gradient and centrifugal direction undermines the slight detour assumption of DiffTriangle and deteriorates its accuracy. This section quantifies this impact of the DG pattern on DiffTriangle, and based on this analysis, we propose DiffTriangle^*—an enhancement to DiffTriangle, which can tolerate both obstacle detour and the interference of multiple anisotropic factors.

Our DiffTriangle^* has the following advantages compared with other algorithms tolerating detoured anchors. Different from RenderedPath [9] assuming a constant number of anchors, DiffTriangle^* can provide higher accuracy, due to its ability (inherited from DiffTriangle) to exploit increased anchor density and tolerate multiple anisotropic factors. PDM [10] is also an algorithm to tolerate

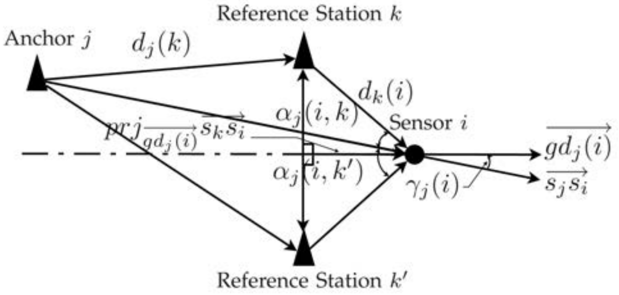

Fig. 11. Ambiguous DiffTriangle in the DG pattern.

both multiple anisotropic factors and obstacle detour. Compared with PDM, DiffTriangle* is a distributed solution with less communication overhead and is robust in all experimental (i.e., rectangular, U-shaped, and O-shaped) networks. PDM, however, degrades severely in the O-shaped deployment fields (see Section 7.5). In contrast to GridVoting [15], DiffTriangle* has higher accuracy, because although GridVoting proposes secured multilateration to drop the lying anchors due to obstacle detour, GridVoting cannot effectively filter moderately detoured anchors, which deteriorates its localization accuracy.

5.1 DG Pattern's Impact on Accuracy of DiffTriangle

We analyze the impact of the DG pattern on distance estimation accuracy of DiffTriangle. We show that the accuracy of DiffTriangle can degrade to $0.8r$ in anisotropic concave networks. This error is not acceptable to many location-dependent applications, since $0.4r$ is the tolerable bound of localization error as argued by [4]. This fact imposes a requirement for more effective anchor-sensor distance estimation algorithms to handle obstacle detour, which is the topic of the next section.

The shortest path between two sensor nodes can strongly deviate from the straight line connecting them, if distorted by obstacles, as illustrated by the exemplified S-shaped network in Fig. 10. This deviation causes a mismatch between field gradient and centrifugal direction, which we summarize as the DG pattern, which undermines the assumption of DiffTriangle and degrades its accuracy. We quantify the degree of obstacle detour by the deviation angle $\gamma_j(i)$.

$$\gamma_j(i) : \text{the deviation angle between gradient direction}$$
$$\overrightarrow{gd_j(i)} \text{ and centrifugal direction } \overrightarrow{s_j s_i}.$$

Based on this definition, we can quantify the impact of obstacle detour on DiffTriangle's accuracy as

$$\pm \sin \alpha_j(i,k) \cdot d_k(i) \cdot \gamma_j(i). \qquad (12)$$

The above quantification is based on the ambiguous DiffTriangle in Fig. 11, which clearly shows the deviation angle $\gamma_j(i)$. Due to the existence of nonzero $\gamma_j(i)$, sensor i cannot tell whether its reference station and the anchor lie on the same side of the gradient $\overrightarrow{gd_j(i)}$ or on opposite sides. This ambiguity is represented in Fig. 11 as the two cases, reference stations k and k'. Caused by this ambiguity of stations k and k', the $\alpha_j(i,k)$ required by DiffTriangle in its Fig. 8 has ambiguous values as represented by $\alpha_j(i,k)$ and $\alpha_j(i,k')$ in Fig. 11. Therefore, (9) for DiffTriangle should be modified to the following equation for ambiguous DiffTriangle.

$$\alpha_j(i,k),\ \alpha_j(i,k') \approx \arccos \left[prj_{\overrightarrow{gd_j(i)}} \overrightarrow{s_k s_i} \ / \ d_k(i) \right] \pm \gamma_j(i).$$

Therefore, the impact of the ambiguous deviation angle $\pm\gamma_j(i)$ on accuracy of DiffTriangle can be quantified as $\pm \sin \alpha_j(i,k) \cdot d_k(i) \cdot \gamma_j(i)$, by assigning $\pm\gamma_j(i)$ to $\Delta\alpha_j(i,k)$ in (10).

Our conclusion is that the accuracy of DiffTriangle in the DG pattern can degrade to $0.4r + 0.4r = 0.8r$, since the accuracy of DiffTriangle in the CG pattern is roughly $0.4r$ and the additional impact of obstacle detour on the accuracy of DiffTriangle is $\pm 0.4r$. This impact $\pm 0.4r$ of obstacle detour is estimated by (12) as $\pm\frac{1}{2} \cdot 1.6r \cdot \frac{\pi}{6} \approx \pm 0.4r$, since in typical concave anisotropic networks, the average station-sensor distance $d_k(i)$ is roughly 2 hops $\approx 1.6r$ and the average deviation angle $\gamma_j(i)$ is $\frac{\pi}{6}$ (see Fig. 10). Moreover, the impact $\pm 0.4r$ of obstacle detour indicates that distance estimates by DiffTriangle maybe either enlarged or diminished.

5.2 Proposed DiffTriangle* and wMultilateration(8)

To tolerate the gradient distortion (or obstacle detour), we present our solution comprising of two parts: *wMultilateration(n)* and $DiffTriangle^*$. wMultilateration(8) is a weighted multilateration [12] method using only the nearest eight anchors.

- wMultilateration(8) deploys the *Nearest(8)* filter at the CG recognizer to reduce the chance to contain detoured anchors in the CG category, assuming the nearest eight anchors are less vulnerable to obstacle detour than farther anchors. We choose the threshold eight since eight is beneficial to mitigate the bad geometry effect of anchors [4].
- wMultilateration(8) uses weighted multilateration [12] to prefer nearby anchors, assuming the expected error of CR pattern is $0.2r$ and that of the CG pattern is $0.4r$.

However, there is a non-negligible chance for the CG category to contain detoured anchors, which degrades the performance of DiffTriangle and deteriorates the localization accuracy.

We therefore propose DiffTriangle*, an enhancement to DiffTriangle algorithm with no additional communication cost and able to generate accurate distance estimates with the presence of gradient distortion (call it "recover the distorted anchors" for short). Therefore, our framework in Fig. 2 adopts DiffTriangle* as a better distance estimator for the CG pattern than DiffTriangle, since it is inevitable for the CG category to contain detoured anchors. Although DiffTriangle* has higher accuracy than DiffTriangle, it needs two reference stations from the CR category. Hence, when there is only one anchor in the CR category, DiffTriangle is a good backup for DiffTriangle*, and when there is none, CrMcs is the last choice.

We present the intuition of DiffTriangle* as follows: DiffTriangle* assumes there are two reference stations for sensor i: *Station* k and *altStation* k^* with $d_k(i) < d_{k^*}(i)$. Its main idea is that if sensor i and station k both use the DiffTriangle algorithm and altStation k^* (as the reference station required by DiffTriangle) to estimate their distances to anchor j, then the two distance estimates $\hat{d}_{4j}(i)|_{k^*}$ and $\hat{d}_{4j}(k)|_{k^*}$ by DiffTriangle may encounter similar distortion effects and thus have similar estimation errors. Because the error in $\hat{d}_{4j}(k)|_{k^*}$ is known from the geometric location of station k, it is possible for $\mathrm{DiffTriangle}^*$ to recover the error in $\hat{d}_{4j}(i)|_{k^*}$ due to obstacle detour by knowing the error in $\hat{d}_{4j}(k)|_{k^*}$. As a summary, $\mathrm{DiffTriangle}^*$ is a revision to DiffTriangle (and thus a second order revision to CrMcs), exploiting the geometric closeness between sensor j and station k and recovering the bias in DiffTriangle's distance estimation due to obstacle detour.

We present the equation of $\mathrm{DiffTriangle}^*$ as follows:

$$\begin{aligned}\hat{d}_{5j}(i)|^{k}_{k^*} &= \hat{d}_{4j}(i)|_{k^*} - \Delta\hat{d}_{4j}(i)|_{k^*}\\ &= \hat{d}_{4j}(i)|_{k^*} - \Delta\hat{d}_{4j}(k)|_{k^*}\cdot\frac{\hat{d}_{3k^*}(i)}{\hat{d}_{3k^*}(k)}.\end{aligned} \tag{13}$$

In the above equation, the sensor i first obtains an unrevised anchor-sensor distance estimate $\hat{d}_{4j}(i)|_{k^*}$ by DiffTriangle algorithm (with altStation k^* as the required reference station). Then, we revise the $\hat{d}_{4j}(i)|_{k^*}$ to $\hat{d}_{5j}(i)|^{k}_{k^*}$ by the estimated error $\Delta\hat{d}_{4j}(i)|_{k^*}$ in $\hat{d}_{4j}(i)|_{k^*}$. The estimated error $\Delta\hat{d}_{4j}(i)|_{k^*}$ is a linear transformation from the estimated error $\Delta\hat{d}_{4j}(k)|_{k^*}$ in the anchor-station distance estimation. The above correction is possible, since the station k and the sensor i are geometrically close to each other and the anchor j's hop field near them thus probably experiences similar distortion with similar deviation angle $\gamma_j(i)\approx\gamma_j(k)$. The aim in (13) of multiplying the revision $\Delta d_{4j}(k)$ by the ratio $\frac{\hat{d}_{3k^*}(i)}{\hat{d}_{3k^*}(k)}$ is to remove the impact of sensor-altStation distance $d_{k^*}(i)$, since our analysis indicates that $d_{k^*}(i)$ linearly interferes the accuracy of DiffTriangle.

We present the analysis on error characteristics of DiffTriangle, showing that $d_{k^*}(i)$ linearly interferes the accuracy of DiffTriangle. According to (10), if sensor i applies DiffTriangle and uses altStation k^* as the reference station, the distance estimation error of DiffTriangle can be estimated by the following equation:

$$\begin{aligned}\Delta\hat{d}_{4j}(i)|_{k^*} \approx{}& \cos\alpha_j(i,k^*)\ \Delta d_{k^*}(i)\\ &+ \sin\alpha_j(i,k^*)\ d_{k^*}(i)\ \Delta\alpha_j(i,k^*).\end{aligned}$$

From the above equation, we know the accuracy of DiffTriangle is mainly affected by two factors: the altStation-sensor distance $d_{k^*}(i)$ and the deviation angle $\gamma_j(i)$ from obstacle detour. According to our analysis in the previous section, the deviation angle $\gamma_j(i)$ directly interferes our estimation of angle $\alpha_j(i,k^*)$, whose impact over $\Delta d_{4j}(i)$ is $\pm\sin\alpha_j(i,k^*)\cdot d_{k^*}(i)\cdot\gamma_j(i)$. The aim of $\mathrm{DiffTriangle}^*$ is to recover this impact from obstacle detour. The other factor affecting accuracy of DiffTriangle is $d_{k^*}(i)$, because 1) there exists a tight bond (linear approximately) between $d_{k^*}(i)$ and the altStation-sensor distance estimation accuracy $\Delta d_{k^*}(i)$ due to the existence of error accumulation if applying the isotropic CrMcs method, and 2) $d_{k^*}(i)$ linearly amplifies the error $\Delta\alpha_j(i,k^*)$ in estimation of angle $\alpha_j(i,k^*)$.

The required error $\Delta\hat{d}_{4j}(k)|_{k^*}$ (or correction) in (13) is provided by station k as follows:

$$\Delta\hat{d}_{4j}(k)|_{k^*} = \hat{d}_{4j}(k)|_{k^*} - d_j(k).$$

In this equation, the station k derives an anchor-station distance estimate $\hat{d}_{4j}(k)|_{k^*}$, by pretending as a normal sensor and applying the DiffTriangle algorithm with altStation k^* as the required reference station. The error in this estimate $\hat{d}_{4j}(k)|_{k^*}$ can be known, since the anchor-station distance $d_j(k)$ can be known from their positions.

An implementation tip of $\mathrm{DiffTriangle}^*$ for reduced communication overhead is that the required correction $\hat{\Delta}d_{4j}(k)$ in (13) provided by the station k can be calculated locally by sensor i, since the station k has already transmitted its Input I to sensor i and the Input II of the station k (that contains the Input I of altStation k^*) is the same with Input II of sensor i (see Fig. 2). Another implementation tip of $\mathrm{DiffTriangle}^*$ for an improved accuracy is to well handle the case that the station k and altStation k^* are roughly of the equal distance to the sensor i. To well handle this, sensor i can make anchors k and k^* take turns to be the station and the altStation. Then, sensor i by applying (13) can obtain two revised distance estimates to anchor j: $\hat{d}_{5j}(i)|^{k}_{k^*}$ and $\hat{d}_{5j}(i)|^{k^*}_{k}$. The final estimate $\hat{d}'_{5j}(i)$ is the weighted average.

$$\hat{d}'_{5j}(i)|^{k}_{k^*}=\frac{1}{\hat{d}_{3k}(i)+\hat{d}_{3k^*}(i)}\Big[\hat{d}_{3k^*}\ \hat{d}_{5j}(i)|^{k}_{k^*} + \hat{d}_{3k}\ \hat{d}_{5j}(i)|^{k^*}_{k}\Big]. \tag{14}$$

6 Pattern-Driven Localization Scheme

We present the pseudocode of our pattern-driven localization algorithm (Algorithm 1), which corresponds to the framework depicted in Fig. 2.

```
Algorithm 1: Localization scheme running on sensor $i$
Input:
1. DGPattern (Input I): the initial set of heard anchors (by
flooding and local broadcast), temporally stored as
DGPattern = Input I = { [$j$, loc$_j$, $h_j(i)$, $|N_j(i)|$, $|N(i)|$] }
2. RefInfo (Input II): the Input I of nearby reference
stations propagated by confined flooding, with
RefInfo = { [$k$, Input I] }, where $k$ is the reference station
id. RefInfo.size indicates the number of heard stations.
Output: location estimate loc$_i$ of sensor i.
begin
    /* Init: System Assembly                              */
    CRThreshold  = 4
    CRRecognizer = HopNoLargerThan(CRThreshold)
    CGRecognizer = Nearest(8)
    CREstimator  = CrMcs                      /* Eq. (5)  */
    RefInfo = HopNoLargerThan(CRThreshold).filter(RefInfo)
    if RefInfo.size > 1 then
        CGEstimator = DiffTriangle*(RefInfo)  /* Eq. (14) */
    else if RefInfo.size is equal to 1 then
        CGEstimator = DiffTriangle(RefInfo)   /* Eq. (7)  */
    else CGEstimator = CrMcs                  /* Eq. (5)  */
    LocEstimator  = wMultilateration          /* Sec. 5.2 */
    /* Phase I: Anchor Classification                     */
    CGPattern = CGRecognizer.filter(DGPattern)
    CRPattern = CRRecognizer.filter(CGPattern)
    /* Phase II: Distance Estimation                      */
    DEs = ∅ ; /* the set of distance estimates            */
    foreach anchor $j \in$ CRPattern do
        DEs += [loc$_j$, CREstimator.estimate(anchor $j$), 0.2r]
    foreach anchor $j \in$ CGPattern do
        DEs += [loc$_j$, CGEstimator.estimate(anchor $j$), 0.4r]
    /* Phase III: Location Estimation                     */
    return loc$_i$ = LocEstimator.estimate(DEs)
end
```

The pseudocode in Algorithm 1 integrates the three algorithms proposed, i.e., CrMcs, DiffTriangle, and DiffTriangle*.

- CrMcs at Line 1 can mitigate the impact of last hop distance for the isotropic anchors in the CR category.
- As the first order revision to CrMcs, DiffTriangle at Line 10 (requiring one reference station) can tolerate the anisotropic anchors in the CG category.
- As the second order revision to CrMcs, DiffTriangle* at Line 8 (requiring two reference stations) can tolerate strongly detoured anchors in the CG category.

An implementation tip for the weighted multilateration [12] at Line 12 is that it is necessary to use multiple start points to perform the multilateration and choose the best solution (with the smallest weighted average residue) as the final location estimate, due to its non-negligible possibility of trap in local minima. One rule of thumb for the reference station filtering (and propagation) at Line 6 is that 1) when the anchors are densely distributed, we only need reference stations within three hops, and 2) when the anchors are sparsely distributed, the threshold can be relaxed to be four hops. For used annotations, please check Fig. 2 and Table 1.

7 Simulation Results

To verify the effectiveness of our pattern-driven localization scheme in tolerating multiple network anisotropic factors, several experiments have been designed and conducted in this section. In these experiments, the accuracy of our scheme has been compared with that of Amorphous [3] and PDM [10] in various network configurations. We choose Amorphous and PDM for comparisons, since 1) Amorphous is a typical isotropic localization algorithm which can be used by an anisotropy-tolerating algorithm to show its power, and 2) PDM is a state-of-the-art algorithm who also declares to handle network anisotropy. We do not compare our algorithm with iMultihop [11], REP [9], or MDS-MAP(P) [8], since they are not the algorithms declared to tolerate multiple anisotropic factors. Our conclusions are: 1) PDM and our scheme have higher distance estimation accuracy and localization accuracy (by tolerating multiple anisotropic factors) than Amorphous, 2) our scheme has lower communication cost than PDM and is more accurate in dense networks, and 3) our scheme is robust in rectangular, O-shaped, and U-shaped networks, but PDM degrades severely in O-shaped networks.

7.1 Evaluation Metrics and Controlled Parameters

We use the metrics below to evaluate the three algorithms (Amorphous, PDM, and our scheme) in experiments.

- Distance estimation error μ_h for h-hop paths:

$$\mu_h = \sum_{\hat{d}_j(i) \in \mathcal{D}_h} \frac{|\hat{d}_j(i) - d_j(i)|}{|\mathcal{D}_h|}, \text{where}$$

$$\mathcal{D}_h = \{\text{distance estimates of all paths with lengths of } h\}.$$

μ_h is useful to observe the accumulation of distance estimation error with the increase of hop count.

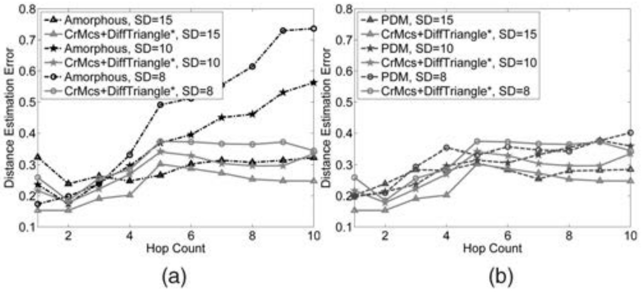

Fig. 12. Distance estimation error varying SD; $\mathrm{DOI} = 0$, Anchor $\mathrm{Num} = 6$, $\mathrm{SDR} = \mathrm{rectangle}\ 10r \times 10r$, $\mathrm{ACR} = 1.5r$. (a) Amorphous and our scheme. (b) PDM and our scheme.

- Overall average distance estimation error μ:

$$\mu = \sum_{\hat{d}_j(i) \in \cup\mathcal{D}_h} \frac{|\hat{d}_j(i) - d_j(i)|}{|\cup \mathcal{D}_h|},$$

  where $\cup\mathcal{D}_h$ is the set of distance estimates of paths with any hops. According to the analysis of multilateration in [13] based on Cramér-Rao bound, localization error should approach average distance error μ, which we will verify in our experiments.
- Average localization error ε: ε is the arithmetic mean of the localization error of all sensors, with $\varepsilon = \sum \varepsilon_i / N$, where N is the total number of sensors and ε_i is the localization error of sensor i with $\varepsilon_i = |\hat{p_i} - p_i|$, which is the geometric distance between the estimated position $\hat{p_i}$ and the true position p_i.

We control the following system parameters and investigate their impact on the above evaluation metrics.

- Sensor Density (SD): The average number of sensors per sensor radio area. $\mathrm{SD} = \overline{|N(i)|}$. (See Table 1).
- Degree of Radio Irregularity (DOI): With the presence of DOI, the possibility of two nodes with distance d to establish a link is $P(d)$ as follows:

$$P(d) = \begin{cases} 1, & \frac{d}{r} < 1 - \mathrm{DOI}, \\ \frac{1}{2\mathrm{DOI}} \left(\frac{d}{r} - 1 \right) + \frac{1}{2}, & 1 - \mathrm{DOI} \leq \frac{d}{r} \leq 1 + \mathrm{DOI}, \\ 0, & \frac{d}{r} > 1 + \mathrm{DOI}. \end{cases}$$

- Shape of Deployment Region (SDR): The rectangular, U-shaped, and O-shaped regions have been studied.
- Anchor Cell Radius (ACR): The average radius of the Voronoi cell dominated by an anchor. ACR $(= \sqrt{\frac{\text{Area of deployment region}}{\pi \cdot \text{Anchor Number}}})$ represents the average distance of a sensor to its nearest reference station.

7.2 Distance Estimation Error versus Sensor Density

This experiment varies the sensor density to study its impact on the accuracy of distance estimation. To isolate the impact of sensor density, we intentionally fix the DOI to 0, the ACR to $1.5r$, and the SDR to rectangular regions $(10r \times 10r)$. The simulation results are plotted in Fig. 12.

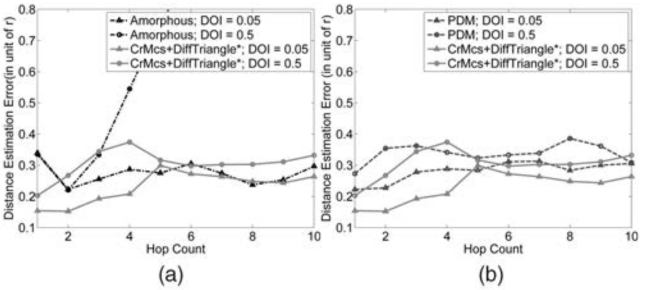

Fig. 13. Distance estimation error varying DOI ratio; $\mathrm{SD} = 15$, $\mathrm{SDR} = \mathrm{rectangle}\, 10r \times 10r$, $\mathrm{ACR} = 1.5r$. (a) Amorphous and our scheme. (b) PDM and our scheme.

We show the advantages of our scheme over Amorphous in Fig. 12a. 1) When SD is as low as 8, Amorphous encounters severe error accumulation as the hop count grows, due to the anisotropy from low sensor density (Section 4.1). However, Amorphous's accuracy is relatively satisfactory in the first three (or four) hops. That is why our scheme approximately treats the first few hops as isotropic regions, where the impact of varying *HopSize* is negligible and the CR pattern approximately applies. Compared with Amorphous, our scheme can suppress the error accumulation and keep the distance estimation error constantly below $0.4r$, which is consistent with our analysis of DiffTriangle's accuracy in Section 4.3. 2) Even in the approximately isotropic region within four hops, our scheme constantly outperforms Amorphous, because CrMcs is a better method to solve the last hop distance problem than Amorphous smoothing.

We compare our scheme with PDM in Fig. 12b. 1) In dense networks ($\mathrm{SD} \geq 15$), the accuracy of PDM is still above $0.2r$, since it neglects the last hop distance problem. In contrast, the accuracy of our scheme can be improved to $0.15r$, when sensor density is 15 and in the first four hops. However, our scheme is only slightly better than PDM when sensor density is 15 and beyond four hops, since the anisotropy (inaccurate $HopSize$) becomes the dominating factor rather the last hop distance. 2) In sparse networks ($\mathrm{SD} \leq 10$), PDM has non-negligible possibility for the failure of its matrix inverse operation and may have to apply the pseudoinverse multiple times. Moreover, sparse networks have higher possibility to be separated and the branch separated from base station may not be localized by PDM. Our scheme, a distributed algorithm, has no such problem.

7.3 Distance Estimation Error Varying DOI Ratio

This experiment investigates the impact of high DOI ratio, such as $\mathrm{DOI} = 0.5$, on distance estimation accuracy. To highlight its impact, we minimize the impact of other anisotropic factors by configuring SD as high as 15 and SDR as rectangle regions. The experimental results are depicted in Fig. 13.

In Fig. 13a, the accuracy of Amorphous degrades severely when $\mathrm{DOI} = 0.5$, since the Kleinrock's equation that Amorphous uses is inaccurate in estimating *HopSize*, assuming a perfect circular communication range. As a comparison, the performance of our scheme however remains stable, when $\mathrm{DOI} = 0.5$, although our scheme also uses the Kleinrock's equation. This is because the estimated *HopSize* is only used by DiffTriangle to estimate the short sensor-station distance and the projection of this distance on the field gradient. As a summary, our DiffTriangle and $\mathrm{DiffTriangle}^*$ method is insensitive to the error in *HopSize* estimation, which gives our scheme the freedom to choose any appropriate *HopSize* estimation algorithm, either online or offline. In Fig. 13b, PDM is also robust against radio irregularity. The accuracy of PDM is slightly lower than our scheme due to the non-negligible last hop distance problem in dense networks.

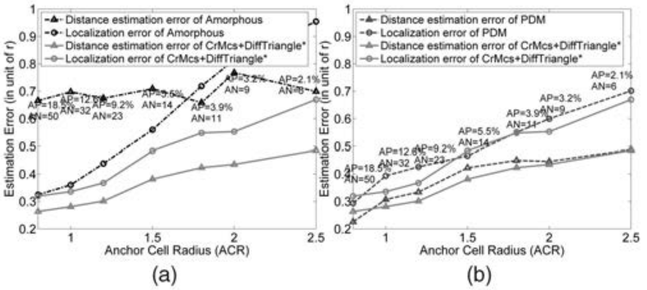

Fig. 14. Distance estimation error and localization error varying ACR; $\mathrm{SD} = 8$, $\mathrm{DOI} = 0.25$, $\mathrm{SDR} = \mathrm{rectangle}$, Sensor Number ≈ 250, Anchor Percentage (AP) $= \frac{\text{Anchor Number (AN)}}{\text{Sensor Number}}$. (a) Amorphous and our scheme. (b) PDM and our scheme.

7.4 Localization Error when Varying ACR

This experiment explores the impact of anchor density (quantified by ACR) over the accuracy of distance estimation and localization. The experiment results are illustrated by Fig. 14, in which we adopt the same localization algorithms wMultilateration(8) for Amorphous, PDM and our scheme to maintain a fair comparison.

In Fig. 14a, the distance estimation accuracy of Amorphous has no change when ACR decreases, since Amorphous has no mechanism exploiting the increased anchor density and optimizing its distance estimation accuracy. In contrast, our scheme and PDM have much smaller distance estimation error below $0.4r$. Although the distance estimation accuracy of Amorphous does not improve with reduced ACR, its localization accuracy constantly improves, if Amorphous adopts wMultilateration(8) for localization. However, Amorphous's localization accuracy deterioration speed (when ACR grows) is much faster than our scheme and PDM. Moreover, for PDM and our scheme, we can find a tight correspondence between average distance estimation error and average localization error, which is consistent with the theoretical analysis in [13] based on Cramér-Rao bound.

The underperformance of Amorphous is because although wMultilateration(8) uses the nearest eight anchors and the weighted multilateration to prefer nearby anchors, it can be quite difficult to precisely capture the error accumulation trend of Amorphous's distance estimation at the location estimator layer. As a comparison, PDM and our scheme choose to suppress the error accumulation mainly at the distance estimator layer by exploiting the dense anchor distribution. For this reason, the localization accuracy of Amorphous (even with wMultilateration(8)) can be above $0.7r$, while that of PDM and our scheme stay below $0.5r$ when ACR is below 2. This performance improvement is important, since the work in [4] has stated that the performance of several location-dependent protocols (like geometric routing) degrades quickly when the localization accuracy is above $0.4r$. Moreover, when anchors

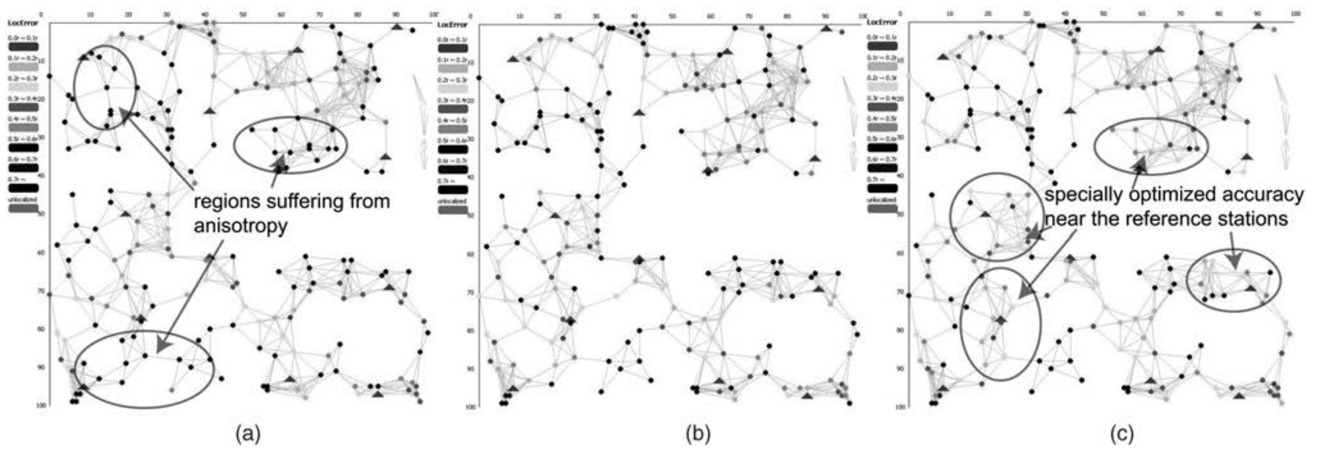
regions suffering from anisotropy
specially optimized accuracy near the reference stations
(a)
(b)
(c)

Fig. 15. Comparison in localization accuracy of Amorphous, PDM, and our scheme; $\mathrm{SD} = 8$, $\mathrm{DOI} = 0.3$, $\mathrm{ACR} = 1.6r$, $\mathrm{r} = 10\ \mathrm{ft}$. (a) Amorphous; $\mathrm{AvgLocError} = 0.72r$. (b) PDM; $\mathrm{AvgLocError} = 0.52r$. (c) Our scheme; $\mathrm{AvgLocError} = 0.46r$.

are densely distributed with $\mathrm{ACR} = 1.5r$, the anchor percentage is merely 5.5 percent, which does not seem to be an unbearable huge investment. The optimized trade-off between anchor percentage and localization accuracy may only be found in specific applications.

7.5 Localization Error when Varying SDR

In this section, we compare the localization accuracy of three algorithms (i.e., Amorphous, PDM, and our scheme) in anisotropic networks with both obstacle detour and the presence of multiple anisotropic factors (radio irregularity and low sensor density). This experiment shows that 1) in the U-shaped regions like Fig. 15, PDM and our scheme can improve the localization accuracy, compared with Amorphous, 2) in the O-shaped regions like Fig. 16, PDM degrades dramatically but our algorithm still functions well. In Figs. 15 and 16, we depict the network topologies as graphs with triangular nodes for anchors, circular nodes for sensors, and links for radio connections. We also label for each node its localization error by color.

In Fig. 15, we compare the localization accuracy of Amorphous, PDM, and our scheme in the U-shaped region (for fairness, all of them use wMultilateration(8) for location estimation). As illustrated, Amorphous suffers from network anisotropy and obstacle detour, with localization accuracy above $0.7r$. If instead applying PDM for anchor-sensor distance estimation in Fig. 15b, the localization accuracy can be improved to around $0.5r$. When we use our scheme (including CrMcs and $\mathrm{DiffTriangle}^*$) to estimate the distances in Fig. 15c, we can achieve similar localization accuracy with PDM. Our scheme, however, has two advantages over PDM.

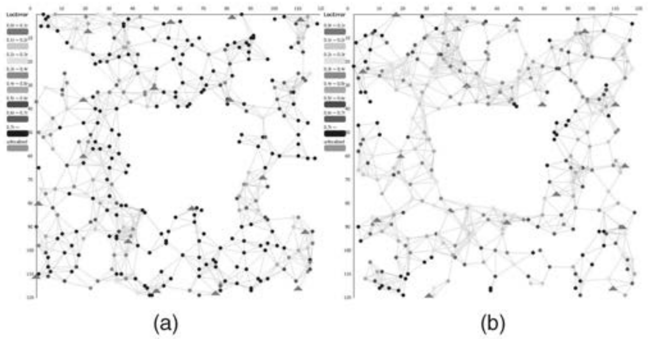

Fig. 16. Localization accuracy of PDM and our scheme in O-shaped region; $\mathrm{SD} = 8$, $\mathrm{DOI} = 0.3$, $\mathrm{ACR} = 1.6r$, $\mathrm{r} = 10\mathrm{ft}$. (a) PDM; $\mathrm{AvgLocError} = 0.87r$. (b) Our scheme; $\mathrm{AvgLocError} = 0.45r$.

1. To achieve this comparable accuracy with PDM, our scheme has simpler communication operation (i.e., the confined flooding) and less communication overhead.

   - The overall communication overhead of PDM is $O(M^2N)$, where M is the number of anchors and N is the number of sensors. PDM first needs the M times flooding initiated by M anchors to make the sensors know their hop counts, whose overhead is $O(MN)$. PDM then needs to collect the hop counts between all pair of anchors to the base station (i.e., M unicasts to the base station), and disseminate the calculated PDM matrix to the entire network (i.e., a flooding of a $M \times M$ matrix), whose communication cost is $O(M^2N)$.
   - The overall communication overhead of our algorithm is $O(MN)$. 1) Besides the M times flooding initiated by M anchors, our algorithm (precisely DiffTriangle and $\mathrm{DiffTriangle}^*$) requires each anchor to propagate its hop counts (to M anchors) to three or four hops' neighborhood by a confined flooding. This is equivalent to 1-2 times flooding of an M-sized vector because a confined flooding may partly overlap with another confined flooding. This has the overhead of $O(MN)$. 2) Another kind of communication needed by our algorithm (precisely CrMcs) is to make every node aware of the number of neighbors with equal and larger hop counts. This needs N times local broadcast of M-sized vector (see paragraph 3 of Section 2), whose overhead is $O(MN)$.

2. Our scheme has specially optimized localization accuracy near each anchor (see Fig. 15c), since sensors use their nearest reference stations to optimize distance estimates. The shorter the distance to reference station, the better the localization accuracy.

Here, comes an interesting feature of our scheme in contrast to PDM: PDM degrades in the O-shaped region, but our scheme still functions well as shown by Fig. 16. According to the simulation in Fig. 16a, the localization accuracy of PDM degrades quickly in the O-shaped region. This degradation is probably because the PDM tries to find an optimum linear transformation between two high-dimensional spaces. (One space is the proximity distances to all anchors and the other is the euclidean distances to all anchors.) This bidirectional linear transformation provides a good approximation in the rectangular, U-shaped, and S-shaped regions, but it is inaccurate in the O-shaped regions. Our localization scheme has no such problem and it still renders satisfying performance in the O-shaped region as in Fig. 16b.

8 Related Work

Our survey of related work is focused on the range-free localization, which is a much cheaper option than range-based and anchor-free localization [17]. This is because the range-free option leverages the radio transmitter already deployed on each sensor and thus eliminates the need for extra per-node devices or additional infrastructure. However, it is more difficult for the range-free localization to achieve high localization accuracy, which is desirable for location-dependent protocols and applications [4]. For a range-free solution, accuracy is the most critical factor in deciding its applicability.

A common feature of pioneering range-free solutions (e.g., DV-Hop [2], Amorphous [3], and MDS-MAP [1]) is the assumption of network isotropy. Hence, their performance degrades severely in testbeds, where anisotropic factors exist (e.g., concave deployment region, sparse and nonuniform sensor distribution, anisotropic terrain condition, and irregular radio pattern). This underperformance has led to extensive research on anisotropy tolerating algorithms. Most of these studies are based on two commonly seen (and also overlapping) assumptions: a fixed number of anchors and the presence of one anisotropic factor (i.e., obstacle detour).

A fixed number of anchors is assumed by MDS-MAP(P) [8] and REP [9]. The algorithms with this assumption share two drawbacks: 1) lack of mechanisms to fully exploit increased anchor density and 2) potential accumulation of error, when the network scale is large. As one example, MDS-MAP(P), following the canonical “divide and conquer” paradigm, splits the network into small overlapping subregions. For each subregion, which is considered to be locally isotropic, MDS-MAP is applied to compute a local map. By merging these local maps, a global map is formed by a coordinate registration procedure. However, this recursive merge operation is sensitive to the noise in local map construction and suffers from error propagation after several iterations [18], especially in large-scale networks.

The presence of only one anisotropic factor (i.e., obstacle detour) is assumed by iMultihop [11] and REP [9]. However, the performance of these algorithms may degrade in practice, due to the inevitable existence of multiple anisotropic factors in real deployment of WSNs. The iMultihop [11] contributes an impressive improvement to the multilateration component of DV-Hop. It is based on the observation that the shortest path from an anchor to a sensor will deviate far away from straight lines when distorted by obstacles, and the distorted distance estimate is always larger than its real value. Therefore, a set of upper bound quadratic inequality constraints can be added to the MMSE objective function of traditional multilateration. However, this assumption of iMultihop that distance estimates are enlarged by obstacle detour may not hold for networks with multiple anisotropic factors. As a summary, the assumption of only one anisotropic factor may weaken the soundness of algorithms designed to tolerate anisotropy.

The problem of range-free localization tolerating network anisotropy can be investigated from another perspective by assuming a varied number of anchors proportional to network scale and the presence of multiple anisotropic factors, rather than a fixed number of anchors and the presence of only one anisotropic factor. The PDM [10] and our scheme both fall into this category. These two solutions are compared as follows: 1) Our scheme is a distributed solution with less communication overhead. The communication cost of our scheme is $O(MN)$, while that of PDM (a centralized algorithm) is $O(M^2N)$, where M is the number of anchors and N is the number of sensors—Section 7.5. 2) Our scheme has consistent performance in various shapes (e.g., rectangular, O-, and U shapes) of sensor deployment fields, but PDM degrades severely in the O-shaped region. 3) Our scheme has higher accuracy than PDM, when sensors are densely distributed, thanks to the ability of CrMcs to handle the last hop distance problem.

9 Conclusion

For accurate localization in networks with multiple anisotropic factors, we propose a pattern-driven localization scheme, applying different distance estimation algorithms for anchors exhibiting different patterns, i.e., the CR, CG, and DG patterns. For the CR pattern, CrMcs is adopted to minimize the impact of last hop distance. For the CG pattern, DiffTriangle is used to tolerate varying *HopSize* and exploiting the rough match between field gradient and centrifugal direction. $\mathrm{DiffTriangle}^*$ provides an enhancement to DiffTriangle to additionally tolerate obstacle detour. For the DG pattern, where the line-of-sight rule no longer holds, the anchors are dropped. Both theoretical analysis and experimental results support the effectiveness of our localization scheme in tolerating multiple network anisotropic factors.

Appendix A
Systematical Error of Amorphous

We analyze the systematical error of Amorphous incurred by the last hop distance, which has been plotted in Fig. 4. To eliminate the impact of inaccurate *HopSize* and isolate the impact of last hop distance on distance estimation of Amorphous, *HopSize* is fixed to be r.

The systematic error E_2 of Amorphous to estimate last hop distance is the difference between expected estimate

$\overline{\hat{d}_{2j}(i)}$ produced by (2) and the real distance value $d_j(i)$. Our analysis is all about how to establish the functional relation from $d_j(i)$ to E_2.

$$E_2 = \overline{\hat{d}_{2j}(i)} - d_j(i). \tag{15}$$

The expected estimate $\overline{\hat{d}_{2j}(i)}$ can be calculated by (16), since there are only three possible values for the hop count $h_j(t)$ of sensor t, which neighbors sensor i. These three possible values are $h_j(i) - 1$, $h_j(i)$, and $h_j(i) + 1$. Moreover, the expected percentage of neighbors of sensor i to have hop count $h_j(i) - 1$, $h_j(i)$, or $h_j(i) + 1$ is equal to the probability, that the hop count $h_j(t)$ is $h_j(i) - 1$, $h_j(i)$, or $h_j(i) + 1$.

$$\begin{aligned}\overline{\hat{d}_{2j}(i)} &= P\left[h_j(t) < h_j(i)\right]\left(h_j(i) - 1\right) + P\left[h_j(t) = h_j(i)\right]\\ &\cdot h_j(i) + P\left[h_j(t) > h_j(i)\right]\left(h_j(i) + 1\right) - 0.5 \text{ with } t \in N(i).\end{aligned} \tag{16}$$

The probability $P[h_j(t) < h_j(i)]$ required by (16) is equal to the division of the area $\mathcal{A}\left(r,\ (h_j(i) - 1)r,\ d_j(i)\right)$ of the intersected region of two disks, by the area $\pi\, r^2$ of sensor i's entire neighbor. The intersecting two disks are sensor i's one hop disk with radius r and anchor j's $h_j(i) - 1$ hop disk, whose radius is approximated to $(h_j(i) - 1)r$. The area of the intersecting region can be calculated, when the distance $d_j(i)$ between the centers of the two disks is known additionally.

$$P\left[h_j(t) < h_j(i)\right] = \frac{\mathcal{A}\left[r,\ (h_j(i) - 1)\, r,\ d_j(i)\right]}{\pi\, r^2}. \tag{17}$$

The probability $P\left[h_j(t) \leq h_j(i)\right]$ is calculated in a similar way with $P\left[h_j(t) < h_j(i)\right]$.

$$P\left[h_j(t) \leq h_j(i)\right] = \frac{\mathcal{A}\left[r,\ h_j(i)\, r,\ d_j(i)\right]}{\pi\, r^2}. \tag{18}$$

The other two probabilities $P\left[h_j(t) = h_j(i)\right]$ and $P\left[h_j(t) > h_j(i)\right]$ in (16) can be calculated as follows:

$$\begin{aligned}&P\left[h_j(t) = h_j(i)\right] = P\left[h_j(t) \leq h_j(i)\right] - P\left[h_j(t) < h_j(i)\right],\\ &P\left[h_j(t) > h_j(i)\right] = 1 - P\left[h_j(t) \leq h_j(i)\right].\end{aligned}$$

The above equations help establish a functional relation from $d_j(i)$ to E_2, with $h_j(i) = \lceil \frac{d_j(i)}{r} \rceil$. This functional relation has been plotted in Fig. 4 and indicates the underperformance of Amorphous in one or two hops. This is consistent with the simulation results in Section 7.2.

Appendix B

Error Characteristics of CrMcs

We show by analysis that the proposed CrMcs can reduce distance estimation error to below $0.2r$ (sensor density > 8), if applied to the CR pattern. The analysis result (i.e., the relation between the error of CrMcs $Dev\,[\hat{d}_{3j}(i)]$ and the sensor density $|N(i)|$) is plotted in Fig. 6. The accuracy of CrMcs mainly depends on two factors:

1. the accuracy of the intersected area $a_j(i)$ by (3), and
2. the accuracy of the radius of $dsk_j(i)$ by (4).

Generally speaking, in dense networks and isotropic networks with accurate *HopSize* estimates, factor 1 dominates, which is directly connected with the last hop distance. However, in sparse networks and anisotropic networks, where the estimation of *HopSize* is inaccurate, factor 2 prevails to have a greater impact. This is why we recommend CrMcs for the CR pattern.

We focus on the impact of factor 1 (i.e., the last hop distance) in the low hop count cases. Driven by this intention, we fix the *HopSize* to r to eliminate the impact of inaccurate *HopSize* and isolate the impact of the last hop distance. We then simplify the expression of $\hat{d}_{3j}(i)$ as

$$\hat{d}_{3j}(i) = \mathcal{A}^{-1}\left[r,\ h_j(i)\cdot r,\ \pi\, r^2 \cdot \frac{|N_j(i)|}{|N(i)|}\right], \tag{19}$$

with $r(1)$ fixed to r and $r(h_j(i))$ fixed to $h_j(i)\cdot r$. In this equation, the only randomized variable that determines $\hat{d}_{3j}(i)$ is $\frac{|N_j(i)|}{|N(i)|}$, which is essentially a Binomial trial testing the percentage of neighbor owning hop count no larger than $h_j(i)$. Hence, the accuracy of CrMcs can be estimated as the average absolute deviation of $\hat{d}_{3j}(i)$:

$$\begin{aligned}Dev\,[\hat{d}_{3j}(i)] = &\sum_{|N_j(i)|\,\in\,[0,\ |N(i)|]} C_{|N(i)|}^{|N_j(i)|} \cdot |\hat{d}_{3j}(i) - d_j(i)| \cdot\\ &\widetilde{P}^{|N_j(i)|} \cdot (1 - \widetilde{P})^{|N(i)\,-\,N_j(i)|},\ where\ \widetilde{P} = \frac{\mathcal{A}[r,\ h_j(i)\cdot r,\ d_j(i)]}{\pi\, r^2}.\end{aligned}$$

Acknowledgments

This work was supported in part by HK PolyU G-U810, CityU project no. 9681001, China NSFC grant no. 60803161 and 973 grant 2009CB320702.

References

[1] Y. Shang, W. Ruml, Y. Zhang, and M.P.J. Fromherz, "Localization from Mere Connectivity," *Proc. ACM MobiHoc*, 2003.

[2] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecomm. Systems*, vol. 22, pp. 267-280, 2003.

[3] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network," *Proc. Second ACM/IEEE Int'l Conf. Information Processing in Sensor Networks (IPSN)*, 2003.

[4] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. ACM MobiCom*, pp. 81-95, 2003.

[5] B. Xiao, H. Chen, and S. Zhou, "Distributed Localization Using a Moving Beacon in Wireless Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 5, pp. 587-600, May 2008.

[6] B. Xiao, L. Chen, Q. Xiao, and M. Li, "Reliable Anchor-Based Sensor Localization in Irregular Areas," *IEEE Trans. Mobile Computing*, vol. 9, no. 1, pp. 60-72, Jan. 2010.

[7] A. Savvides, C.-C. Han, and M.B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM MobiCom*, pp. 166-179, 2001.

[8] Y. Shang and W. Ruml, "Improved MDS-Based Localization," *Proc. IEEE INFOCOM*, pp. 2640-2651, 2004.

[9] M. Li and Y. Liu, "Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes," *Proc. ACM MobiCom*, 2007.

[10] H. Lim and J.C. Hou, "Localization for Anisotropic Sensor Networks," *Proc. IEEE INFOCOM*, pp. 138-149, 2005.

[11] C. Wang and L. Xiao, "Locating Sensors in Concave Areas," *Proc. IEEE INFOCOM*, pp. 1-12, 2006.

[12] W.H. Foy, "Position-Location Solutions by Taylor-Series Estimation," *IEEE Trans. Aerospace and Electronic Systems,* vol. 12, no. 2 pp. 187-194, Mar. 1976.

[13] A. Savvides, W.L. Garber, S. Adlakha, R.L. Moses, and M.B. Srivastava, "On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks," *Proc. Second ACM/IEEE Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2003.

[14] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-Hoc Sensor Networks Using Multidimensional Scaling," *Proc. IEEE INFOCOM,* 2004.

[15] D. Liu, P. Ning, and W.K. Du, "Attack-Resistant Location Estimation in Sensor Networks," *Proc. Fourth ACM/IEEE Int'l Conf. Information Processing in Sensor Networks (IPSN)*, pp. 99-106, 2005.

[16] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six Is a Magic Number," *Proc. Nat'l Telecomm. Conf.,* pp. 431-435, 1978.

[17] R. Stoleru, T. He, and J.A. Stankovic, "Range-Free Localization," *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks,* vol. 30, pp. 3-31, Springer, 2007.

[18] L. Meertens and S. Fitzpatrick, "The Distributed Construction of a Global Coordinate System in a Network of Static Computational Nodes from Inter-Node Distances," Kestrel Inst. Technical Report KES.U.04.04, 2004.

Qingjun Xiao received the BSc degree in computer science from the Nanjing University of Posts and Telecommunications, China, in 2003 and the MSc degree in computer science from Shanghai JiaoTong University, China, in 2007. Now, he is working toward the PhD degree in the Department of Computing at Hong Kong Polytechnic University. His research interest currently is focused on distributed algorithms for wireless sensor networks.

Bin Xiao received the BSc and MSc degrees in electronics engineering from Fudan University, China, in 1997 and 2000, respectively, and the PhD degree in computer science from the University of Texas at Dallas in 2003. Now, he is an assistant professor in the Department of Computing of Hong Kong Polytechnic University. His research interests include distributed computing systems, data storage, and security in wired and wireless computer networks. He is a member of the IEEE.

Jiannong Cao received the BSc degree in computer science from Nanjing University, China, and the MSc and PhD degrees in computer science from Washington State University, Pullman. He is currently a professor and associate head in the Department of Computing at Hong Kong Polytechnic University. His research interests include mobile and pervasive computing, computer networking, parallel and distributed computing, and fault tolerance. He has published more than 250 technical papers in the above areas. He is a senior member of the China Computer Federation, the IEEE, the IEEE Computer Society, and the IEEE Communication Society, and a member of the ACM. He has served as an associate editor and a member of editorial boards of several international journals, including the *IEEE Transactions on Parallel and Distributed Systems*, *Pervasive and Mobile Computing*, and *Wireless Communications and Mobile Computing*.

Jianping Wang received the BSc and MSc degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the PhD degree in computer science from the University of Texas at Dallas in 2003. She is currently an assistant professor in the Department of Computer Science, City University of Hong Kong. Her research interests include wireless networks, optical networks, and network coding. She is a member of the IEEE.