Localization of Wireless Sensor Networks in the Wild: Pursuit of Ranging Quality

Jizhong Zhao, *Member, IEEE,* Wei Xi, *Student Member, IEEE,* Yuan He *Member, IEEE,* Yunhao Liu, *Senior Member, IEEE,* XiangYang Li, *Senior Member, IEEE,* Lufeng Mo, *Member, IEEE,* Zheng Yang, *Member, IEEE,*

Abstract—Localization is a fundamental issue of wireless sensor networks that has been extensively studied in the literature. Our real-world experience from GreenOrbs, a sensor network system deployed in a forest, shows that localization in the wild remains very challenging due to various interfering factors. In this paper we propose CDL, a **C**ombined and **D**ifferentiated **L**ocalization approach for localization that exploits the strength of range-free approaches and range-based approaches using RSSI. A critical observation is that ranging quality greatly impacts the overall localization accuracy. To achieve a better ranging quality, our method CDL incorporates virtual-hop localization, local filtration, and ranging-quality aware calibration. We have implemented and evaluated CDL by extensive real-world experiments in GreenOrbs and large-scale simulations. Our experimental and simulation results demonstrate that CDL outperforms current state-of-art localization approaches with a more accurate, and consistent performance. For example, the average location error using CDL in GreenOrbs system is 2.9m, while the previous best method SISR has an average error of 4.6m.

Index Terms—Localization, Wireless Sensor Network, RSSI, Ranging Quality.

1 Introduction

Localization is crucial for many services provided by wireless sensor networks (WSNs), which has received substantive attention in recent years. The Global Positioning System (GPS) are popular localization schemes, but usually fail to function indoors [11], under the ground [10], or in forests with dense canopies [14]. Range-based approaches measure the Euclidean distances among the nodes with various ranging techniques [16], [20], [24]. They are either expensive with respect to hardware cost, or susceptible to environmental noises and dynamics [23]. Range-free approaches perform localization by relying only on network connectivity measurements. However, localization results by range-free approaches are typically imprecise and easily affected by node density.

This work is motivated by the need for accurate location information in GreenOrbs [14], a large-scale sensor network system deployed in a forest. An indispensable element in various GreenOrbs applications is the location information of sensor nodes for purposes such as fire risk evaluation, canopy closure estimates, microclimate observation, and search and rescue in the wild. Our real-world experiences of GreenOrbs reveal that localization in the wild remains very challenging, in spite of great efforts and results developed in the literature. The challenges come from various aspects. First, non-uniform deployment of sensor nodes could affect the effectiveness of range-free localization. On the other hand, for range-based localization, the received signal strength indicators (RSSIs) used for estimating distances are highly irregular, dynamic, and asymmetric between pairs of nodes. To make it even worse, the complex terrain and obstacles in the forest easily affect RSSI-based range measurements, thus incurring undesired but ubiquitous errors.

Ranging based localization techniques often produces better localization than range-free techniques. Ranging quality determines the overall localization accuracy. Bearing this in mind, recently proposed approaches focused more on error control and management. Some of those methods enhance the localization accuracy by deliberately reducing the contribution of error-prone nodes to the localization process [13]. Other schemes are to identify large ranging errors and outliers relying on topological or geometric properties of a network [7], [27].

Ranging quality indeed includes two aspects. One of them refers to the location accuracy of the reference nodes. The other concerns the accuracy of range measurements. Both aspects play important roles on the accuracy of localization. Most of the recently proposed techniques address only one aspect, thus failing to achieve satisfactory accuracy.

To address these challenges and limitations, we propose **CDL**, a **C**ombined and **D**ifferentiated **L**ocalization approach. CDL inherits the advantages of both range-free and range-based methods. It starts from a coarse-grained localization achieved by method such as DV-

- *Wei Xi, Jizhong Zhao, Lufeng Mo are with Xi'An Jiaotong University.*
- *Yuan He and Zheng Yang are with the Tsinghua National Laboratory for Information Science and Technology, Tsinghua University.*
- *Yunhao Liu is with the School of Software and TNLIST, Tsinghua University, and also with the Hong Kong University of Science and Technology.*
- *XiangYang Li is with Department of Computer Science, Illinois Institute of Technology.*
- *The preliminary result was published at ACM SenSys 2010.*

hop, and then it keeps improving the ranging quality and localization accuracy iteratively throughout the localization process. The contributions of this work are summarized as follows.

1) We propose a range-free scheme called virtual-hop localization, which makes full use of local information to mitigate the non-uniform node distribution problem. Using virtual-hop, the initial estimated locations are more accurate than those output by other range-free schemes.
2) To improve the ranging quality, we design two local filtration techniques, namely *neighborhood hop-count matching* and *neighborhood sequence matching*, to find nodes with better location accuracy. The filtered good nodes can be used to improve the location accuracy of neighboring nodes.
3) Using the good nodes to calibrate the bad ones, we employ the weighted robust estimation to emphasize contributions of the best range measurements, eliminate the interfering outliers, and suppress the impact of ranges in-between.
4) We implement CDL in GreenOrbs system with more than 300 sensor nodes deployed in a forest and evaluate it with extensive experiments and large-scale simulations. Our experimental and simulation results demonstrate that CDL outperforms existing approaches with high accuracy, efficiency, and consistent performance. For example, the average location error using CDL in GreenOrbs system is 2.9m, while the previous best method SISR has an average error of 4.6m.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 presents real-world observations on GreenOrbs. The design of CDL is elaborated in Section 4, followed by performance evaluation in Section 5. We conclude the paper in Section 6.

2 Related Work

The existing work on localization falls into two main categories: range-based and range-free localization.

Range-free approaches, such as Centroid [2], APIT [5], and DV-HOP [17], mainly rely on connectivity measurements (for example hop-count) from landmarks to the other nodes. Since the quality of localization is easily affected by node density and network conditions, range-free approaches typically provide imprecise estimation of node locations. Range-based approaches measure the Euclidean distances among the nodes with certain ranging techniques and locate the nodes using geometric methods, such as TOA [1], TDOA [18], [20], and AOA [16]. All those approaches require extra hardware support.

RSSI-based range measurements are easy-to-implement and popular in practice. Empirical models of signal propagation are constructed to convert RSSI to distance [21]. The accuracy of such conversions, however, is sensitive to channel noise, interference, and multipath effects. Besides, when there are a limited number of landmarks, range-based approaches have to undergo iterative calculation processes to locate all the nodes, suffering significant accumulative errors [13].

More recent proposals mainly focus on the issue of error control and management [12], [26]. J. Liu *et al.* [13] propose iterative localization with error management. Only a portion of nodes are selected into localization, based on their relative contribution to the localization accuracy, so as to avoid error accumulation during the iterations. Similarly, H.T. Kung *et al.* [8] propose to assign different weights to range measurements with different nodes and adopt a robust statistical technique to tolerate outliers of range measurements [7].

A range-free approach beyond connectivity is proposed in [27]. The *signature distance* is proposed as a measure of the Euclidean distance between a pair of nodes. In order to address the issue of non-uniform deployment, the authors further propose *regulated signature distance* (RSD), which takes node density into account. Based on the comparison among nodes' neighbor sequences, RSD is quantified. This approach needs to be integrated with a certain existing localization approach to function.

Differing with most of the existing approaches, CDL is a combination of range-free and range-based schemes. It can independently localize a WSN. CDL addresses the issue of non-uniform deployment with virtual-hop localization (Subsection 4.1). Utilizing the information of estimated node locations, RSSI readings, and network connectivity, CDL filters good nodes from bad ones with two techniques (Subsection 4.2), namely neighborhood hop-count matching and neighborhood sequence matching. CDL pursues better ranging quality (namely more accurate reference locations and more accurate ranging) throughout the localization process. This is the most significant characteristic of CDL that distinguishes it from existing approaches.

For ease of presentation, we use the terms "ranging" and "range measurement", "location" and "coordinates", interchangeably throughout the rest of this paper.

3 Preliminary and Design Motivation

3.1 GreenOrbs

GreenOrbs is an ongoing research project that aims at building long-term large-scale WSN systems in the forest. It adopts TelosB motes with MSP430 processor and CC2420 radio. The software running on the nodes is developed based on TinyOS 2.1. There are 330 nodes in a deployment area of about $40,000m^2$. The majority of Greenorbs nodes should be deployed where environmental information is required by forestry applications. The rest are used to improve network connectivity.

The collected data can be utilized to support a wide variety of applications, *e.g.* distance-dependent competition measurement for predicting growth of individual trees, light detection and ranging to characterize forest

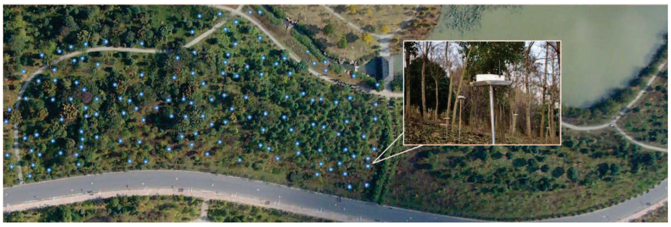


Fig. 1: GreenOrbs deployment in the campus woodland

stand condition, and percentage estimation of ground area vertically shaded by overhead foliage. These applications generally require accurate coordinates of sensor nodes' locations to provide high-quality information of the forest [9], [14], [25].

This work is carried out in GreenOrbs. The ground-truth coordinates of the nodes are measured using an EDM (Electronic Distance Measuring Device) [3]. The measurement process is hence laborious and time-consuming. So far we have succeeded in measuring the coordinates of 100 nodes, as shown in Fig. 1. The observations and experiments in this paper are then mainly conducted using those 100 nodes. The other 230 nodes, although deployed in an adjacent area, are not shown in the figure.

3.2 Observations

As shown in Fig. 1, most sensor nodes are under dense tree cover, where GPS usually does not work [1]. Even in areas with less dense tree cover, our experience shows that the errors produced by a portable GPS device (compared to an EDM) are often about 15m. Thus locating nodes basically comes down to in-network localization. This subsection presents real-world observations on GreenOrbs, which illustrate that a single approach, whether it is range-based or range-free, has limitations in locating a number of nodes in the wild.

3.2.1 Non-uniform Deployment

Driven by forestry applications, GreenOrbs deploys more sensor nodes in regions with diverse or uneven vegetation to provide fine-grained information of the monitored area. Such a rule leads to non-uniform deployment of sensor nodes, as we can see from Fig. 1. Specifically, some nodes have more than 20 neighbors, while some nodes have less than 5 neighbors. The shortest distance is 5m and the longest is around 108m. Range-free localization in a non-uniform deployment often incurs large errors.

3.2.2 Irregularity of RSSI

Besides the non-uniform deployment problem, complex terrain and obstacles (*e.g.* shrubs and tree trunks) also affect signal propagation in the forest. Fig. 2 plots the RSSI between node pairs in GreenOrbs at a certain time. It also includes a curve, which shows the mapping between RSSI and the distance based on the log normal shadowing model, see Equation (1).

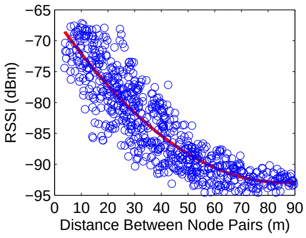


Fig. 2: RSSI of different node pairs

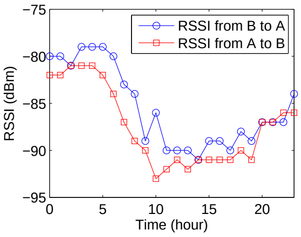


Fig. 3: RSSI between nodes A and B over time

$$PL(d) = PL(d_0) - 10 \times \eta \times \log(\frac{d}{d_0}) + X_\sigma \qquad (1)$$

where $PL(d)$ denotes the reduction in received signal strength after propagating through a distance d, $PL(d_0)$ stands for the path loss at a short reference distance d_0, η is the path loss factor (also named signal propagation constant), and X_σ is a random environment noise following $X \sim N(0, \sigma_{X^2})$ reported in [19].

We can see that the real distances between node pairs differ greatly from the model-based estimations. Though the mapping between the RSSI and the distance is actually very uncertain, RSSI still offers useful information. In most cases, a stronger RSSI corresponds to a shorter distance, as is also observed in [4], [27].

3.2.3 Asymmetry and Dynamics of RSSI

Fig. 3 shows the RSSI of two directed links AB and BA between two nodes A and B in GreenOrbs over time. The distance between A and B is 41.27 meters. We can see that the RSSI between two nodes is asymmetric. Two pairwise links often have unequal RSSI. Moreover, RSSI is often susceptible to environmental factors, such as humidity and temperature. The RSSI over a directed link also fluctuates over time.

In summary, we have the following important observations on GreenOrbs. First, the sensor nodes are deployed with diverse densities in different regions, causing the non-uniform distribution problem. Second, RSSI is very unstable and sensitive to various environmental factors. The uncertainty of RSSI is hard to model in practice, therefore, RSSI-based range measurements exhibit quite diverse errors. To make matters even worse, typically only large ranging errors can be detected or tolerated by the existing approaches.

4 CDL Design

We consider locating a network of wireless nodes on a two dimensional plane by using the connectivity information and RSSI readings. A few nodes, which know their own coordinates once they are deployed, are used as landmarks. The design of CDL mainly consists of *virtual-hop localization*, *local filtration*, and *ranging-quality aware calibration*. Fig. 4 illustrates the CDL workflow.

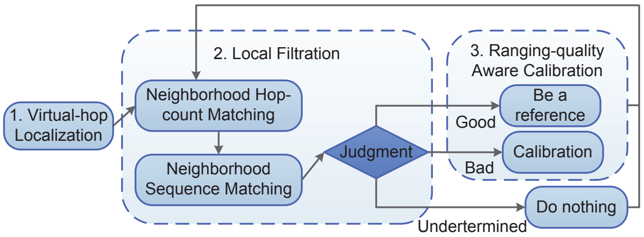

1. Virtual-hop Localization
2. Local Filtration
Neighborhood Hop-count Matching
Neighborhood Sequence Matching
Judgment
3. Ranging-quality Aware Calibration
Good
Be a reference
Bad
Calibration
Undertermined
Do nothing

Fig. 4: The workflow of CDL

Virtual-hop localization initially estimates node locations using a range-free method. In order to approximate the distances from each node to the landmarks, we let each node count the virtual-hops instead of DV-hops, compensating particularly for the errors caused by the non-uniform deployment problem.

Subsequently, CDL executes an iterative process of *filtration* and *calibration*. In each filtration step, CDL uses two filtering methods to identify good nodes whose location accuracy is already satisfactory. *Neighborhood hop-count matching* filters the bad nodes by verifying a node's hop-counts to its neighbors. Furthermore, *neighborhood sequence matching* distinguishes good nodes from bad ones by contrasting two sequences on each node. Each sequence sorts a node's neighbors using a particular metric, such as RSSI and estimated distance.

Those identified good nodes are regarded as references and used to calibrate the location of bad ones. Links with different ranging quality are given different weights. Outliers in range measurements are tolerated using robust estimation.

In the next three subsections, we elaborate on the design of the above three phases respectively.

4.1 Virtual-Hop Localization

For the first phase of CDL, virtual-hop localization initially computes node locations. This is an enhanced version of hop-count based localization. Compared to the DV-hop scheme, virtual-hop particularly addresses the issue of non-uniform deployment. Based on the output of virtual-hop localization, the subsequent localization processes in CDL (filtration and calibration) are expected to achieve higher accuracy and efficiency of iteration.

4.1.1 Weakness of Range-free localization algorithm

As analyzed in [15], there is a theoretical limitation on range-free localization algorithm that is based only on connectivity. Suppose sensor nodes are randomly distributed in the monitoring area. Each sensor can be regarded as a node in a graph, so that two nodes are connected by an edge if and only if they can communicate with each other in one hop, *i.e.* they are less than the distance r from each other. It is possible to move a sensor node over nonzero distance without changing the set of its one-hop neighbors. The original and moved locations of nodes are indistinguishable from the point of view of the network connectivity. The average Euclidean distance between its original location and a moved location that does not changing the network connectivity gives a lower bound on the expected resolution achievable.

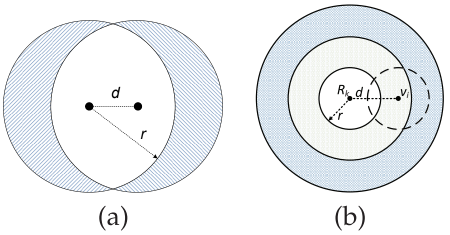


Fig. 5: Intuition of virtual-Hop distance: (a) cumulative distribution of node distances, (b) relationship among neighbors with different hop counts.

As shown in Fig. 5(a), a sensor node can be moved distance d without changing the connectivity, if there is no sensor in the shaded area.

Nagpal *et al.* [15] have claimed that $r\pi/4n_{local}$ is the expected lower bound for the error in any range-free localization algorithm in static sensor networks where n_{local} is the connectivity degree, and the nodes only use the connectivity information of the seeds within their first-hop neighborhoods.

DV-hop is one of the common range-free localization approaches that utilize connectivity information to estimate node locations. Every node counts its hop counts to landmarks. The distance between a node and a landmark is calculated as the product of the hop count between them and the per-hop distance, which is a predetermined constant for all the nodes. The location of a node is calculated by using Least Squares Estimation. However, nodes with the same hop often have quite different distances to landmarks. Fig. 6 shows some nodes that are within three hops away from the landmark. For example, nodes v_a and v_b are both two hops away from landmark R_k, while v_a is closer to landmark than v_b. A constant value of per-hop distance for every node often causes errors on distance calculation from a node to landmarks. As a result, the localization accuracy of DV-hop is far from satisfactory.

4.1.2 Virtual-hop

Since traditional hop-count based technology doesn't differentiate two distances with the same hop counts, we propose a metric *virtual-hop-count*, V_{jk}, to represent the distance between an ordinary node v_j and a landmark R_k. Among the nodes with the same hop count to R_k, node closer to R_k should have a smaller V_{jk}. For ease of presentation, Table 1 lists the symbols and notations used in this paper. Each node v_j computes its V_{jk} by

$$V_{jk} = \frac{1}{|P_{jk}|} \sum_{v_i \in P_{jk}} V_{ik} + L_{jk} \qquad (2)$$

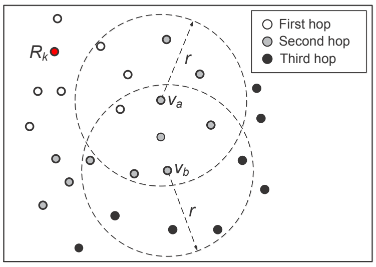


Fig. 6: The same hop counts have different distances

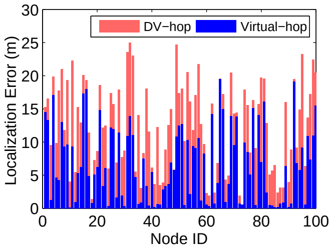


Fig. 7: Virtual-hop VS DV-hop

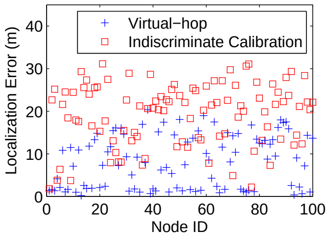


Fig. 8: Indiscriminate Calibration

TABLE 1: Symbols and notations

Symbol	Definition
h_{ij}	hop count from node v_i to node v_j
V_{jk}	virtual-hop-count from landmark R_k to node v_j
$\Re_j$	$\{v_i \mid h_{ij} = 1\}$
P_{jk}	$\{v_i \mid h_{ij} = 1$ and $h_{ki} < h_{kj}\}$
N_{jk}	$\{v_i \mid h_{ij} = 1$ and $h_{ki} > h_{kj}\}$
ζ_{jk}	$\min\{\lvert P_{ik}\rvert \mid v_i$ is in $N_{jk}\}$
φ_{jk}	$\min\{\lvert N_{ik}\rvert \mid v_i$ is in $P_{jk}\}$

where

$$L_{jk} = \begin{cases} \dfrac{|N_{jk}|}{|N_{jk}| + \zeta_{jk} - 1}, & |N_{jk}| > 0 \\ \dfrac{\varphi_{jk}}{|P_{jk}| + \varphi_{jk} - 1}, & |N_{jk}| = 0 \end{cases}$$

V_{jk} consists of two parts: The first part is the average virtual-hop-count of node v_j's previous-hop neighbors. The second part is the last virtual-hop-count, that is, the incremental virtual-hop-count from v_j's previous-hop neighbors to v_j, denoted by L_{jk}. Here, a node v_j's previous-hop neighbor is defined as a neighboring node whose hop count to landmark R_k is just one hop less than v_j, (denoted by P_{jk} in Table 1). v_j's next-hop neighbor is defined as a neighboring node whose hop count is just one hop more than v_j (denoted by N_{jk} in Table 1).

We now explain the intuition behind our definition of virtual-hop-count using probability analysis. Fig. 5(b) shows the relationship among neighbors with different hop counts. The concentric circles separately denote the location boundary of one-hop, two-hops and three-hops neighbors of landmark R_k. The dashed circle denotes the communication range of v_i who is a two-hops neighbor of R_k. The intersection, denoted as $A(P_{ik})$, of dashed circle and small circle (centered at R_k) is the region where v_i' previous-hop neighbors locate. The intersection, denoted as $A(N_{ik})$, of the dashed circle and the big circle centered at R_k is the region where v_i's next-hop neighbors could locate.

For any node v_i, as long as the distance between it and landmark R_k (denoted by d) satisfies $r < d < 2r$, it has two hops to R_k. In this case, the maximum residual of two distances with the same hop count is close to r. For virtual-hop, such two nodes have different virtual-hop-counts. For ease of explanation, we assume connectivity degree is n_{local} and calculate the residual of node v_i's last virtual-hop-count to R_k denoted by L_{ik} defined in Equation 2. The closer v_i is to R_k, the larger the area of $A(P_{ik})$, and smaller L_{ik} is, while DV-hop has a constant hop count. The maximum value of L_{ik} is close to 1. The minimum value of L_{ik} is close to $\frac{1}{\zeta_{ik}} = 1/[\frac{n_{local}}{\pi r^2} \int_0^Y (\sqrt{4r^2 - y^2} - 2r + \sqrt{r^2 - y})dy] \approx \pi/1.4n_{local}$. The upper bound for expected ranging error of DV-hop is r, while the bound for virtual-hop is $\pi r/1.4n_{local}$. Therefore, virtual-hop can reduce both the upper bound and average of localization error when n_{local} is greater than 3.

Let $\mathcal{R}$ be the set of landmarks in the sensor network whose exact positions are known in advance. Let ρ_{tk} be the Euclidean distance between landmarks R_t and R_k. The per-virtual-hop distance, denoted as $\tilde{d_k}$, regarding landmark R_k is calculated by

$$\tilde{d_k} = \frac{\sum_{R_t \in \mathcal{R}} \rho_{tk}}{\sum_{R_t \in \mathcal{R}} V_{tk}} \tag{3}$$

Each node v_j without known location then estimates its distance, denoted as ρ_{jk} , to each landmark R_k by

$$\rho_{jk} = \tilde{d_k} \cdot V_{jk} \tag{4}$$

After calculating the distances to landmarks, each node computes its coordinates based on trilateration using Least Square Estimation (LSE), which is similar to DV-hop.

4.1.3 *Localization Accuracy of Virtual-hop*

We carry out an experiment using the data from GreenOrbs to compare virtual-hop localization with DV-hop, which includes 100 ordinary nodes and 4 landmarks. The experimental result is shown in Fig. 7. We can see virtual-hop outperforms DV-hop remarkably. The performance gain of using virtual-hop varies much among different nodes.

By fully exploiting the connectivity information of the local neighborhood, virtual-hop-counts finely characterize the non-uniform distribution properties with more reasonable hop counting. Nevertheless, it is worth noticing that there are still sizable errors ($> 5m$) at many nodes. Those nodes with sizable location errors should be identified and calibrated. We will present the solutions in the next two subsections. Without causing confusion, hereafter we use “estimated coordinates” to denote the node coordinates before filtration.

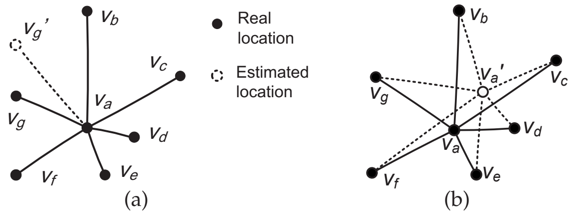


Fig. 9: ADM reflects the localization error of a node

Given the estimated coordinates, the iterative process of filtration and calibration further enhances localization accuracy. This involves the following two design criteria: First, filtration must identify as many *good nodes* with high localization accuracy as possible to facilitate calibration. Second, a *good node* is likely to have both *good* and *bad links*. Only the *good links* (with small ranging errors) should dominate calibration, while the impact of the *bad links* must be restrained. Filtration addresses the first criterion, while calibration resolves the second.

4.2 Local Filtration

Filtration consists of two steps: neighborhood hop-count matching and neighborhood sequence matching. Neighborhood hop-count matching identifies the bad nodes with apparently wrong coordinates according to the residual between the real hop counts and estimated hop counts. Neighborhood sequence matching distinguish good nodes from bad ones according to the matching degree between RSSI sequence and distance sequence.

4.2.1 Large Error of Model-based Filtration

Filtration is very important in CDL. In order to illustrate its significance, we carry out an experiment to examine the efficacy of location calibration without differentiating good nodes and bad nodes. We call this straightforward model-based calibration *indiscriminate calibration*. Using such calibration, every node's location is adjusted directly based on the distances to neighbors converted from RSSI, using the log-normal shadowing model.

Fig. 8 compares the localization errors of nodes before and after *indiscriminate calibration*. In this experiment, we set the parameters as $\eta = 3.3$, $X_\sigma = 6$. Surprisingly, we find the output to be even worse than before. Model-based filtration is infeasible, considering the estimated localization error and irregularity of RSSI.

Based on the information available, there are two ways to estimate the distances between two nodes, for example v_i and its neighbor v_j. One way is to calculate the distance based on their estimated coordinates, denoted by d'_{ij}. The other converts the RSSI from v_j to v_i into a distance (tentatively named RSSI-distance) based on the log-normal shadowing model, denoted by d_{ij}. Ideally, we expect $d_{ij} = d'_{ij}$. Due to the errors of estimated coordinates and the error from the log-normal shadowing model, however, there is often some difference between them. By summing up $|d_{ij} - d'_{ij}|$ corresponding to every neighbor v_j, we can measure the *Aggregated Degree* of Mismatches (ADM) of v_j.

ADM actually reflects the error of a node's estimated location. For example in Fig. 9 (a), v_a is a *good node* (whose estimated location is close to its real location) with six neighbors. Among them only v_g is a bad node. Let v'_g denote its estimated location. Clearly the ADM of v_a is mainly caused by v'_g. In Fig. 9 (b), v_a is a bad node with six good neighbors. The link to every neighbor contributes to the ADM of v_a. By comparing these two figures, we can see the ADM of a bad node is typically higher than that of a good one. Thus we may distinguish good nodes from bad ones by contrasting their ADMs.

4.2.2 Neighborhood Hop-Count Matching

To quantify ADM, each node takes neighborhood hop-count matching as the first step to identify whether it is a *good node* based on local connectivity information. Note that hop-count is indeed a rough estimation of the distance between two nodes. If a node's hop-counts to its neighbors greatly mismatches the distances calculated using the nodes' estimated coordinates, *w.h.p.* the local node's coordinates will have a large error. We use v_i as an example to illustrate the matching procedure.

First, every node exchanges the estimated coordinates with its 2-hop neighborhood. Second, after received the estimated coordinates of v_j, v_i estimates the distance between them, denoted by d'_{ij}. Third, for each node v_j within its 2-hop neighborhood, v_i estimates the hop-count to v_j as $h'_{ij} = \lceil d'_{ij}/r \rceil$, where r is the communication range. Fourth, v_i computes its ratio of matched hop-counts within its 2-hop neighborhood v_j as follows:

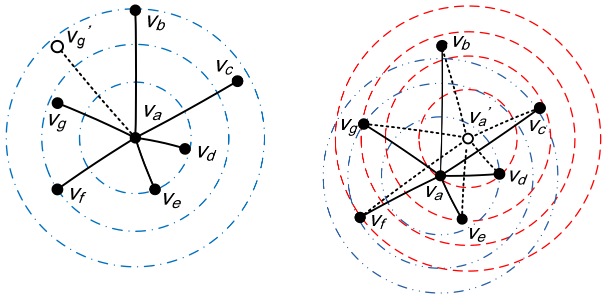
$$H_{ij} = \begin{cases} 0 & (h_{ij} \neq h'_{ij}) \\ 1 & (h_{ij} = h'_{ij}) \end{cases} \tag{5}$$

$$\tilde{r}_i = \frac{1}{n} \sum_{j=1}^{n} H_{ij} \tag{6}$$

$$\overline{r}_i = \frac{1}{n+1} (\sum_{j=1}^{n} \tilde{r}_j + \tilde{r}_i) \tag{7}$$

where h_{ij} denotes the hop count from v_i to v_j and n is the number of its 2-hop neighbors of v_i. $\overline{r}_i$ denotes the mean matched ratio in the neighborhood of v_i. If $\tilde{r}_i < \overline{r}_i$, v_i regards itself as a bad node, which has an apparent error in its estimated coordinates. Otherwise, the role of node v_i is left undetermined for further filtration.

Hop-counts actually offer relatively limited information to filtration. As a result, neighborhood hop-count matching only identifies a small portion of bad nodes with apparently wrong coordinates. In order to ensure that all the sifted good nodes do have satisfactory location accuracy, we need to further filter bad nodes. In the next subsections, we illustrate our scheme of neighborhood sequence matching.



N_a	B	C	D	E	F	G
S_a	6	5	1	2	4	3
S_a'	6	4	1	2	3	5

(a)

N_a	B	C	D	E	F	G
S_a	6	5	1	2	4	3
S_a'	4	2	1	3	6	5

(b)

Fig. 10: Neighborhood sequence matching

4.2.3 Neighborhood Sequence Matching

Though model-based straightforward filtration is infeasible, RSSI still offers useful information. Generally, the RSSI between two nodes decreases monotonically as the distance increases observed from the RSSI readings in Fig. 2. Based on this observation, we propose a filtration scheme called *neighborhood sequence matching*.

First, v_a sorts its neighbors in descending order with regard to the RSSI from them, generating a sequence number for each neighbor. By mapping the sequence numbers into v_a, we get the first sequence called *RSSI sequence*. Let S_a denote it, as illustrated in Fig. 10.

Second, according to the estimated coordinates, v_a sorts its neighbors in the ascending order with regard to the estimated distance to them, generating the second sequence called *distance sequence*. Let S_a' denote it.

In environment without noises, S_a and S_a' should be identical. If there is significant mismatch between them, it indicates a large error in the node's estimated coordinates. We use the same examples as that in Fig. 9 to illustrate the above idea. As shown in Fig. 10 (a), there is not a significant mismatch between S_a and S_a' in this case. Comparatively in Fig. 10 (b), there appears to be significant mismatch between S_a and S_a'.

Now that the difference between S_a and S_a' is caused by following categories of reasons: the location estimation errors, the irregularity of RSSI between v_a and its neighbors, and the log norm shadowing model for estimating distance using RSSI.

Since the location estimation error is analyzed before, we discuss the influence of the irregularity of RSSI. From the Fig. 2, we can think that RSSI still satisfies the property that it decreases with the increase of the distance between two neighboring nodes.

The next step is to quantify the distance between RSSI sequence and distance sequence to distinguish good nodes from bad ones. In order to improve the filtration performance, we need to suppress the influence of the irregularity of RSSI first.

The cosine distance is a measure of similarity between two vectors by finding the cosine of the angle between them. It is considered to be used to measure the similarity between sequences S_a and S_a'. Given two vectors of attributes, the cosine distance is represented using a dot product and magnitude as following:

$$\begin{aligned} CosDist &= \frac{a_1 a_1' + a_2 a_2' + ... + a_n a_n'}{\sqrt{a_1^2 + a_2^2 + ... + a_n^2}\sqrt{a_1'^2 + a_2'^2 + ... + a_n'^2}} \\ &= \frac{a_1 a_1' + a_2 a_2' + ... + a_n a_n'}{1^2 + 2^2 + ... + n^2} \end{aligned} \tag{8}$$

In Equation (8), a_1, a_2, ..., an are the sequence numbers in S_a while a_1', a_2', ..., a_n' are the sequence numbers in S_a'. These two sequences are actually two different permutations of 1, 2, ..., n. Thus they are two equal sets. The cosine distance filtration reduces the influence of RSSI irregularity. For example, RSSI sequence S_a is $\{6, 5, 1, 2, 4, 3\}$, and distance sequence S_a' is $\{6, 4, 1, 2, 3, 5\}$ as shown in Fig. 10 (a) $CosDist_a$ is equal to 0.967. As the irregularity, S_a occurs local flips in the nodes with similar distance such as v_d and v_e, or v_f and v_g. It may become $\{6, 5, 2, 1, 3, 4\}$, then $CosDist_a$ becomes 0.978 which is close to the theoretical value. The cosine filtration distance has good fault tolerance to suppress the influence of RSSI irregularity. Upper bound of $CosDist$ is 1, lower bound is $\frac{1\cdot n+2(n-1)+3(n-2)+...+n\cdot 1}{1^2+2^2+...+n^2} = \frac{n+2}{2n+1}$, which is not less than 0.5.

However, when a good node has some bad neighbors with large location errors, the cosine distance between two sequences of a good node does not apparently differ from that of a bad node. To deal with this issue, we introduce the LCS (longest common subsequence) length ratio δ_a. Let n denote the number of v_a's neighbors. Then δ_a denotes the ratio of the length of the LCS between S_a and S_a' to n. It is easy to see that the LCS length ratio of a good node is higher than that of a bad node.

The LCS length ratio δ_a is error-tolerant to interference of bad neighboring nodes with large location estimation errors. The boundary of δ is between 0 and 1.

We define the *matching degree* M_i between the *RSSI sequence* and *distance sequence* as follows.

$$M_i = \delta_i \cdot CosDist_i \tag{9}$$

Clearly M_i is a better metric to distinguish good nodes from bad nodes. When a small portion of RSSI readings have relatively large errors, or a good node has some bad neighbors with large location errors, the matching degree cannot be influenced too much.

We use the same trace as that in Fig. 7 to calculate the matching degree of all the nodes after initial localization. The results are plotted in Fig. 11. Nodes of a matching degree over 0.6 have location errors of less than 4 meters. We regard them as good nodes. Nodes of less than 0.4 degree have location errors over 5 meters. We regard them as bad nodes. The other nodes have matching degrees between 0.4 and 0.6, but their location errors vary from 0.1 to 12 meters. The excessive number of bad neighbors with large location estimation errors or

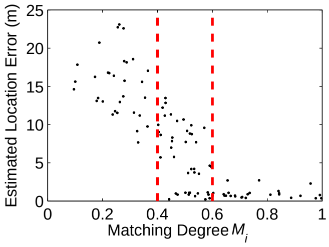


Fig. 11: Filtration result of virtual-hop

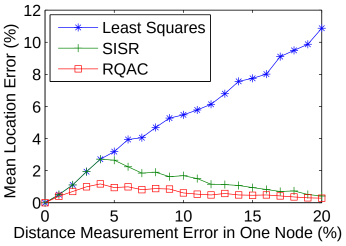


Fig. 12: Localization performance

bad RSSI measurements causes some good nodes with relatively low matching degree. It is by far too hard to decide whether they are good or bad. Thus we tentatively set them as undetermined nodes.

For ease of expression later, we use G_i as a mark of node v_i. $G_i = 0$, $G_i = 0.5$ and $G_i = 1$ mean v_i is a bad node, an undetermined node, and a good node.

$$G_i = \begin{cases} 0 & M_i < \tau_l \\ 0.5 & \tau_l < M_i < \tau_u \\ 1 & M_i > \tau_u \end{cases} \tag{10}$$

Here $\tau_l = 0.4$ and $\tau_u = 0.6$ are two empirical parameters, called the lower matching threshold and the upper matching threshold. One can increase both thresholds to execute stricter filtration. One can also decrease both thresholds to allow more nodes to contribute as good nodes in the calibration process. The tradeoff in the threshold settings could be an interesting issue to study. We leave it for future work.

4.3 Ranging-Quality Aware Calibration

4.3.1 Motivation of RQAC Approach

Given the range measurements between bad node v_i and its good neighbors, the estimation of v_i's location usually works by minimizing an objective function, denoted by f^*, over node pairs (i, j), which is denoted by

$$f^* = \sum_j g(i, j) \tag{11}$$

where $g(i, j)$ takes different forms with different approaches. We use RSSI for calibration, which adjusts the node locations so as to minimize (9).

When LSE is used,

$$g(i, j) = (l_{ij} - d_{ij})^2 \tag{12}$$

where l_{ij} denotes the distance estimated by LSE and d_{ij} denotes the RSSI range measurement between v_i and its neighbor v_j based on the log-normal shadowing model. The problem with LSE is that it does not differentiate between nodes and links. LSE leads to error diffusion where a bad link will seriously affect good links. It suffers great errors when outliers are present in locations or range measurements.

Snap-Inducing Shaped Residuals (SISR) [8] outperforms LSE by assigning different weights to the range measurements with different neighbors.

$$g(i, j) = \begin{cases} \alpha(l_{ij} - d_{ij})^2 & |l_{ij} - d_{ij}| < \lambda \\ \ln(l_{ij} - d_{ij} - u) - v & \text{otherwise} \end{cases} \tag{13}$$

where α, λ, u and v are constant parameters. Once a node is identified as either a good or bad node, its contribution to the calibration is fixed.

SISR actually prefers the uneven situations where the majority of range measurements are accurate. It is proposed to cope with the presence that small amounts of ranging measurements have large non-Gaussian errors. It is inefficient in GreenOrbs System where ranging errors are not uneven as shown in Section 4.1.

To address the limitations of LSE and SISR, our scheme, called range-quality aware calibration (RQAC), adopts the weighted robust estimation technique.

4.3.2 RQAC Estimator

As the set of undetermined nodes include both good and bad, we only use good nodes as references and do not include any undetermined nodes in the calibration. From the viewpoint of v_i, the ranging quality of its neighbor v_j is simultaneously determined by two factors: the location accuracy of v_j, and the ranging error over the link from v_j to v_i. RQAC estimates the ranging quality of a good node v_j with its good neighbors as follows.

$$\tilde{\omega}_j = \sum_{k=1}^{|\Re_j|} \omega'_{jk} \cdot \lfloor G_k \rfloor \tag{14}$$

$$\omega'_{jk} = \begin{cases} 1 & |l_{jk} - d_{jk}| < \theta \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where θ is a pre-configured parameter, $\lfloor G_k \rfloor$ ensures that each good node only communicates with its good neighbors to estimates its ranging quality. The weight of good node v_j in calibrating bad nodes is defined as a normalized value of $\tilde{\omega}_j$.

$$\omega_j = \frac{\tilde{\omega}_j}{\sum_{k=1}^{|\Re_j|} \tilde{\omega}_k} \tag{16}$$

We can see that good nodes of different ranging quality have different weights. A good node has a relatively high weight if its estimated location is highly accurate and the ranging quality of all its links is good. Otherwise, the weight of the good node will be relatively low. The objective function of RQAC is defined as follows:

$$g(i, j) = \begin{cases} \alpha\omega_j(l_{ij} - d_{ij})^2 & |l_{ij} - d_{ij}| < \varepsilon \\ \ln(|l_{ij} - d_{ij}| - \varepsilon + 1) & |l_{ij} - d_{ij}| \geq \varepsilon \end{cases} \tag{17}$$

Note that ω_j and $|l_{ij} - d_{ij}|$ thus jointly denote the ranging quality from v_j to v_i and α is constant parameter.

As we can see from Equation (17), range measurements to v_i are divided into two classes according to their ranging quality. The range measurements with errors less than ε contribute more to the calibration process by taking the quadratic form of $|l_{ij} - d_{ij}|$. For a range measurement with an error not less than ε, its contribution is suppressed by taking the logarithmic form of $|l_{ij} - d_{ij}|$. Moreover, range measurements in the same class are also differentiated from each other, by taking the weights of reference nodes (ε_j) into account. In this way, RQAC respects the contributions of the best range measurements, eliminates the interference of outliers, and suppresses the contributions from the ranges in-between.

4.3.3 Analysis of RQAC: An Illustrative Simulation

As for the parameter setting in RQAC, a small θ expresses a conservative calibration strategy. Only a small fraction of the best range measurements receives enough respect, which results in highly accurate calibration but likely more rounds of iterations. A large θ expresses an optimistic calibration strategy. Many good range measurements make contributions, such as increasing the efficiency of iterations but likely introducing new errors. Getting an appropriate ε is also important to RQAC. Basically, a smaller ε results in more accurate calibration, and also increases the possibility of falling into the local minimum. In contrast, a lager ε may cause RQAC to degrade to ordinary LSE. In our work, we get θ and ε from the empirical results of our experiments.

In the simulation, we placed 30 nodes on a plot of 100 * 100, where $\sigma = 3$ and $\varepsilon = 5$. Fig. 12 is an illustrative experiment comparing localization performance of least-squares, SISR and RQAC under exactly erroneous link of one node. The mean error of least-squares grows along with the input error, while the results of SISR and RQAC go up a little and then decreases into place. That is because the influence of bad link is weakened by the estimator when measurement error exceeds a certain level. Furthermore, the suppression result of SISR is not as significant as RQAC. For RQAC, if the link errors are less than the certain level, their contributions may also be different from each other. The introduction of ω can treat distinctively for different nodes with different location accuracy and link quality.

The RQAC estimator is based on robust statistics. Robust statistics methods [6] is tools for statistics problems in which underlying assumptions are inexact. It is well known that the least squares (LS) error estimates can be arbitrarily wrong when outliers are present in the data. The estimation can be altered without bound by an extremely noisy outlier. In contrast, the median estimator is not as susceptible to such polluting data, and is considered a robust estimator.

The RQAC is a weighted method. Each node has different ranging quality with different weight values. The ranging quality ω_i of node v_i is decreased with the increasing of link errors. For SISR estimator, the function will sustain quadratic growth when link error below a threshold. For RQAC estimator, the growth trend is restrained by decreasing ranging quality ω.

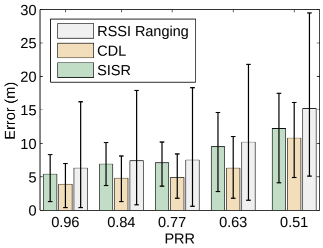


Fig. 13: Error at different PRR

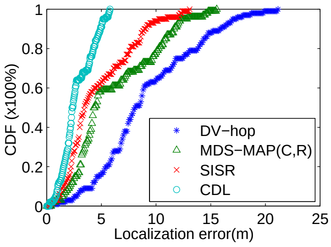


Fig. 14: Overall localization results

5 Performance Evaluation

We have implemented CDL with GreenOrbs. The performance of CDL and other three existing localization approaches, namely DV-hop [17], MDS-MAP(C,R) [22], and SISR [8], is evaluated through real experiments and large-scale simulations.

5.1 Experiments on Real Outdoor System

Corresponding to the deployment map in Fig. 1, Fig. 15 plots the 100 GreenOrbs nodes in a rectangular region. Four nodes positioned near the border of the deployment area are selected as landmarks. In our experiments, we use a globally synchronized duty-cycling mechanism and the CTP protocol to collect data from the nodes. There are two kinds of data: one is sensing data (*i.e.* temperature, humidity, illumination, *etc*), and the other is networking data (*i.e.* neighbor node IDs, RSSI, routing path, *etc*).

The localization experiments are implemented based on the collected data traces in an offline manner. According to the list of neighbor node IDs, the hop counts from landmarks to each node can be calculated. Then localization result of DV-hop algorithm is obtained. According to connectivity information and RSSI readings, the localization performance of MDS-MAP(C,R), SISR, and CDL can be worked out. In Subsections 5.1.1 and 5.1.2, experiments of CDL and SISR are executed for six iterations.

5.1.1 Impact of unreliable wireless links

Fig. 13 compares the RSSI-ranging error and localization performance of CDL and SISR using one month data of GreenOrbs. During that month, the environmental factors, such as temperature, humidity, and wind power, changed frequently. As a result, the system experienced fluctuating packet reception rate (PRR). We use PRR as the indicator of wireless link quality, the change of which actually reflects the impact of environmental dynamics.

RSSI ranging error is the residual between real distance and estimated distance by RSSI ranging based on Equation (1), which usually increases as the PRR shrinks. CDL outperforms SISR under all the five PRR. The local filtration and ranging-quality aware calibration of CDL tend to select the nodes and links with good ranging quality. This tendency appears to have more apparent

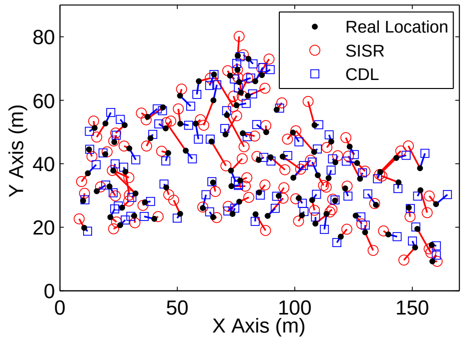


Fig. 15: Localization results of SISR and CDL

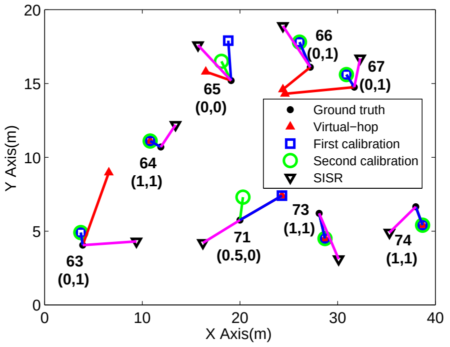


Fig. 16: Impact of Multipath Effect

effect when the quality of wireless links becomes diverse, suppressing the negative impact of unreliable wireless links on the ranging results.

When PRR decreases to 51%, the average RSSI-ranging error increases to more than 15m, and the minimum error is close to 30m. That is because the changes of environmental factors affect the reliability of wireless links. These outside interferences cause irregular RSSI readings and PRR degradation. A high PRR indicates relatively regular RSSI readings and stable environment.

In order to have a good performance, we select the data in consecutive duty cycles with PRR above 96% for all the rest experiments. We set the parameters as $\eta = 3.3$, $X_\sigma = 6$, according to the empirical results [19].

5.1.2 *Comparison among Approaches*

Fig. 14 plots the cumulative distribution of the localization errors using the four approaches. It is easy to see that SISR performs better than DV-hop and MDS-MAP(C,R). Thus we only compare the results of SISR and CDL in Fig. 15.

Fig. 15 shows that for almost all the nodes, CDL achieves higher accuracy than SISR. A detailed explanation of the results can be found in Fig. 14.

Using CDL, 100% of the nodes have errors of less than 7 meters, while 65% of them have errors of less than 3 meters. Using SISR, at most 70% of nodes have errors of less than 7 meters and at most 35% of nodes have errors of less than 3 meters. It is also interesting to see that CDL achieves the most consistent performance among the four approaches. The average localization errors of the four techniques are 8.7m, 5.9m, 4.6m and 2.9m.

From Fig. 14, we can see the performance of DV-hop is the worst. Actually, we observe in the experimental results that many different nodes are estimated to the same locations by DV-hop, because they have the same hop-counts to the landmarks, but their real locations are far each other.

Another interesting finding is that SISR and MDS-MAP perform similarly. In other words, a node with a large error in MDS-MAP usually has a large error in SISR too. Moreover, due to the “snap-in” behavior of SISR, it is able to suppress the negative impact of noisy range measurements. SISR therefore achieves slightly better accuracy than MDS-MAP.

5.1.3 *Impact of Multipath Effect*

In the forest, the complex terrain and obstacles may cause multipath effect. As shown in Fig. 1, there are many big trees in the left center of the deployment area, which makes surrounding nodes’ RSSI readings irregular. The big tree trunks obstruct communication among the nodes on opposite sides (*e.g.* nodes 65 and 73). The RSSI readings among them are weakened. At the same time, the trunks reflect signal from the same side (*e.g.* node 66 and 67). The RSSI readings among the nodes are strengthened.

Fig. 16 compares the localization results of CDL and SISR in this area. The numbers above the braces are the node IDs, the numbers in the braces indicate the local filtration results in two iterations. 0, 0.5 and 1 indicate the node judged to be a bad node, an undetermined node, and a good node, respectively.

Intuitively judging based on the result in Fig. 16, the localization result of CDL approaches the real location of a node step by step, even when some of the wireless links are dominated by multipath effect. That is because that CDL combines range-free and RSSI-based techniques to play their respective advantages. In order to give more insights on how CDL achieves this goal, we use nodes 65, 66, 67, 71 and 73 as a typical example and show their localization process in Table 2.

Table 2 shows the RSSI-ranging error and its impact in the iterative process. The first row is the node ID of sender and the first column is the node ID of receiver. In each cell, the first item is the RSSI-ranging error. Positive number indicates the RSSI is weakened, and negative number indicates the RSSI is strengthened. The second and third items respectively show the error of each link used in the first and second calibration.

Since node 65 is always judged to be bad node, the RSSI-ranging from it is not used in the calibration steps. After the first calibration, nodes 66 and 67 are judged to be good nodes. They are used to calibrate nodes 65 and 71. Although there are large RSSI-ranging errors between nodes 66 and 71, the ranging error of node 66 in calibration is not big. Ranging-Quality Aware Calibration

TABLE 2: RSSI-ranging error and its impact (m)

	65	66	67	71	73
65		-4.81, 0, -1.43	-2.77, 0, -0.92	10.34, 0, 0	10.87, 1.87, 1.65
66	-5.43, 0, 0		-4.85, 0, 0	8.49, 0, 0	5.21, 1.46, 0
67	-2.51, 0, 0	-4.22, 0, 0		1.63, 0, 0	-0.93, -0.97, 0
71	11.74, 0, 0	11.55, 0, 2.04	2.37, 0, 0.93		-2.14, 0, -0.79
73	11.26, 0, 0	5.37, 0, 0	-0.65, 0, 0	-2.78, 0, 0	

(RQAC) can limit the influence of large ranging error with the weighted robust estimator.

Overall, the node with bad ranging quality will either be judged to be a bad node during local filtration or be suppressed with respect to its weight in calibration by RQAC. Thus, CDL can deal with the local multipath effect well.

5.1.4 Interaction of Three Phases

CDL mainly consists of three phases: *virtual-hop localization, local filtration, and ranging-quality aware calibration*. Fig. 17 shows how these methods interact with each other. To simplify the notations, we use the numbers 1, 2, 3 to represent the three phases. Then there are four kinds of combinations: $(2 \cup 3)$, $(1 \cup 2)$, $(1 \cup 3)$, and $(1 \cup 2 \cup 3)$. The different bars indicate the mean localization errors of different combinations.

For $(2 \cup 3)$, we use DV-hop instead of virtual-hop to initialize locations of ordinary nodes. This combination has large localization errors. That is because DV-hop initializes many nodes' locations to be far away from their real locations. Then good nodes and bad nodes, good links and bad links, cannot be easily differentiated. It has serious impact upon the local filtration and ranging-quality aware calibration, and finally reduces the localization accuracy. From this we can see the great importance of virtual-hop in CDL, which provides accurate initial localization.

For $(1 \cup 2)$, we use Least Squares Estimate instead of RQAC for calibration. This combination has higher accuracy than $(2 \cup 3)$. That is because virtual-hop localization provides accurate localization for most nodes. In this situation, nodes can be properly distinguished as good nodes or bad ones. Meanwhile, it has larger maximum error than $(1 \cup 3)$. That is because Least Squares Estimate algorithm leads to error propagation when there are some bad links. It indicates that it is meaningful and beneficial to differentiate the ranging quality of different links in the calibration phase.

For $(1 \cup 3)$, we use RQAC to directly calibrate each node without local filtration. This combination has larger minimum error than $(1 \cup 2)$. Without distinguishing good nodes from bad nodes, it's difficult to evaluate the ranging quality due to the interference of bad nodes. Without appropriate differentiation, the good nodes' locations are also calibrated by their neighbors, as reduces the localization accuracy of good nodes. It indicates that the negative impact of bad nodes may be serious and cannot be neglected. In order to achieve highly accurate localization in the end, we need to filter the bad nodes first before entering the calibration phase.

5.2 Simulation on Large Scale Networks

Besides the above experiments, we further carry out extensive simulations to evaluate the performance of CDL. We examine the location accuracy of CDL by tuning a series of parameters such as network topology, connectivity degree and the relative ranging errors. The results of DV-hop, MDS-MAP(C,R), and SISR are presented as well. The simulations run on Matlab, including 1000 ordinary nodes in a square region and 6 landmarks around. We run all the simulations on a Windows 7 PC with an Intel i5 2.53GHz processor and two core memories size of 2 Gbytes.

In the simulation setting of Section 5.2.1, each node has 10 to 12 neighbors for uniform distribution, and has 3 to 15 neighbors for non-uniform distribution. In Section 5.2.2 and 5.2.3, nodes are randomly distributed in a square region. Each node's RSSI readings from its neighboring nodes are assigned with values based on the log normal shadowing model with random noise, to be more close to the real fact. Two nodes are connected with a link in the network, if the RSSI between them is greater than -87dBm (the receiving sensitivity of CC2420 radio). In this way, the network topology is generated.

5.2.1 Impact of Network Topology

Virtual-hop is a range-free localization which utilizes the connectivity information to locate sensor nodes. We examine the performance of virtual-hop in both scenarios with uniform distribution and non-uniform distribution. DV-hop algorithm takes 43 seconds to run in either uniform or non-uniform distribution simulations. Virtual-hop takes 57 seconds to run in uniform distribution simulation and 58 seconds to run in non-uniform distribution simulation.

Fig. 18 compares the performance of virtual-hop and DV-hop localization approaches in both scenarios. The results indicate that the non-uniform deployment of nodes does build up the average localization errors for both two approaches. It is worth noticing that even virtual-hop localization in the non-uniform deployment is more accurate than the performance of DV-hop localization in the uniform deployment. DV-hop doesn't differentiate between two distances with the same hop count to landmark, while virtual-hop-count assigns small values to the near nodes.

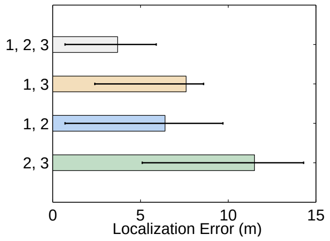


Fig. 17: Interaction of three phases

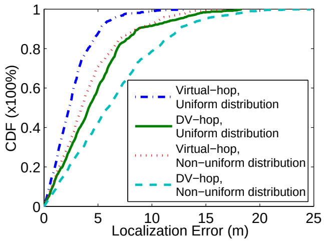
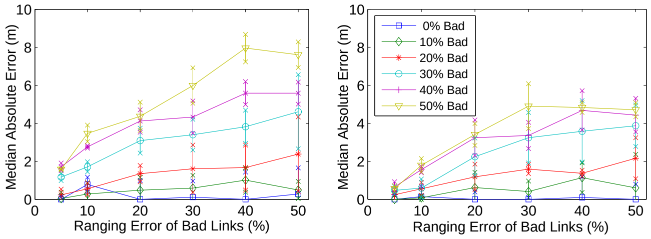


Fig. 18: Impact of Node Distribution



(a) SISR

(b) CDL

Fig. 19: Comparison of localization errors

5.2.2 *Impact of Ranging Error*

Considering the ubiquitous ranging errors in the wild, the robustness of a localization approach against such interfering factors is the last but not least metric we want to evaluate. For this purpose, n_{local} is set to 12.

We use two parameters to control the degree of ranging errors. The first one is the percentage of bad links which is respectively set at 0%, 10%, 20%, 30%, 40%, and 50%. The other parameter is the relative ranging error. We assume in the simulations that the links on a node are either all good or all bad. The relative ranging error of a link conforms to a Gaussian distribution $N(\mu_{bad}, 0.2\mu_{bad})$, where μ_{bad} denotes the average of relative ranging error and set at 0%, 10%, 20%, 30%, 40%, and 50%, respectively. Meanwhile, we assume the links are asymmetric. CDL and SISR are executed for six iterations. SISR runs for 282 min, and CDL runs for 213 min.

Fig. 19 plots the mean localization errors of SISR, and CDL under different settings. SISR is specifically well when the percentage of bad links is less than 30%. The mean localization errors are less than 2m due to the "snap-in" behavior of SISR. Its performance seriously degrades when the percentage of bad links gets above 30%, in accordance with our analysis in Section 4.3.

Compared to SISR, CDL has even better performance. When all the links are good, its localization errors reach near zero. Even when there are 50% bad links, CDL still performs robustly enough. The mean localization error is around 5m. This simulation shows the remarkable advantages of CDL in extremely complex environments.

5.2.3 *Overhead Analysis*

Though cost is not the first concern of localization, we analyze the communication cost and time complexity in each phase of CDL. Let m denote the number of beacon nodes and k denote the average node degree.

In virtual-hop localization, landmarks flood their coordinates to all the other nodes. The communication cost for each ordinary node is $O(m)$. A node exchanges relevant information with its one-hop neighbors to estimate virtual hop-counts. The communication cost is $O(k)$. Finally, landmarks flood their per-virtual-hop distance and the cost is $O(m)$. The overall communication cost for each node in virtual-hop localization is thus to $O(k)$. Node v_i computes its virtual-hop-count to landmark R_j based on the average of its previous neighbors' virtual-hop-counts, so the time complexity is $O(|P_{i,j}|)$. Since a node uses LSE to compute its coordinate, the time complexity is $O((m-1)^3)$.

In local filtration, the communication cost of a node is mainly incurred by information exchange with its one-hop/two-hop neighbors. Thus the communication cost in this phase is $O(k^2)$. The algorithm, called Longest Common Subsequence Length, takes $O(k^2)$ time to compute, and the algorithm Cosine Distance takes $O(k)$ time to yield the output.

In RQAC, all cost is incurred by local computation and thus ignorable, compared to the communication costs in the previous two phases. Each bad node v_i uses the robust estimator to calibrate its location, and the running time of that procedure is $O((n-1)^3)$, where n is the number of v_i' good neighbors.

6 Conclusion

Localization has been extensively studied by both practicers and theoreticians over the past decade. Many practical challenges exist for the state-of-the-art schemes, especially when it comes to real-world WSNs in complex environments. In this work, we share our real-world experience, design, and evaluation of sensor nodes localization with GreenOrbs, a system deployed in a forest. Our design, called CDL, applies a step-by-step process to pursue the best possible localization quality. We have implemented CDL and carried out extensive experiments and simulations. The results demonstrate that CDL outperforms existing approaches with higher accuracy, efficiency, and consistent performance in the wild. Though this work may not be generalized to every possible case, we hope that the community could benefit from our understanding of the practical challenges of localization in large scale WSN deployed in wild.

7 Acknowledgments

This work was supported in part by NSFC Major Program under Grant 61190110, China 973 Program under Grants 2011CB302705 and 2012CB316200, NSFC under Grants 61170213 and 61171067, China 863 Program under Grant 2011AA010100, NSF CNS-0832120 and CNS-1035894, China Postdoctoral Science Foundation under Grant 2011M500019.

References

[1] "Global positioning System. Theory and Practice." *Springer, Wien (Austria), 1993, 347 p., ISBN 3-211-82477-4, Price DM 79.00. ISBN 0-387-82477-4 (USA).*, vol. 1, 1993.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, pp. 28–34, 2000.

[3] S. Crouter, P. SCHNEIDER, M. Karabulut, and D. BASSETT JR, "Validity of 10 electronic pedometers for measuring steps, distance, and energy cost," *Medicine & Science in Sports & Exercise*, vol. 35, no. 8, pp. 1455–1460, 2003.

[4] Z. Guo, Y. Guo, F. Hong, X. Yang, Y. He, Y. Feng, and Y. Liu, "Perpendicular intersection: Locating wireless sensors with mobile beacon," in *2008 Real-Time Systems Symposium*. IEEE, 2008, pp. 93–102.

[5] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of ACM MobiCom*. ACM, 2003, pp. 81–95.

[6] P. Huber and E. Ronchetti, *Robust statistics*. John Wiley & Sons Inc, 2009.

[7] L. Jian, Z. Yang, and Y. Liu, "Beyond triangle inequality: sifting noisy and outlier distance measurements for localization," in *Proceedings of IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[8] H. Kung, C. Lin, T. Lin, and D. Vlah, "Localization with snap-inducing shaped residuals (SISR): Coping with errors in measurement," in *Proceedings of ACM MobiCom*. ACM, 2009, pp. 333–344.

[9] T. Ledermann, "Evaluating the performance of semi-distance-independent competition indices in predicting the basal area growth of individual trees," *Canadian Journal of Forest Research*, vol. 40, no. 4, pp. 796–805, 2010.

[10] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, p. 10, 2009.

[11] ——, "Rendered path: Range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[12] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proceedings of ACM/IEEE IPSN*. IEEE, 2005, pp. 91–98.

[13] J. Liu, Y. Zhang, and F. Zhao, "Robust distributed node localization with error management," in *Proceedings of ACM MobiHoc*. ACM, 2006, pp. 250–261.

[14] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy closure estimates with greenorbs: Sustainable sensing in the forest," in *Proceedings of ACM SenSys*. ACM, 2009, pp. 99–112.

[15] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proceedings of ACM/IEEE IPSN*. Springer, 2003, pp. 553–553.

[16] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of IEEE INFOCOM*. IEEE, 2003, pp. 1734–1743.

[17] ——, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, 2003.

[18] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: a high accuracy acoustic ranging system using COTS mobile devices," in *Proceedings of ACM SenSys*. ACM, 2007, pp. 59–72.

[19] T. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996.

[20] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of ACM MobiCom*. ACM, 2001, pp. 166–179.

[21] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.

[22] ——, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.

[23] L. Vandendorpe, "Multitone spread spectrum communication systems in a multipath Rician fading channel," *Mobile Communications Advanced Systems and Components*, pp. 440–451, 1994.

[24] X. Wang, J. Luo, Y. Liu, S. Li, and D. Dong, "Component-based localization in sparse wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 540–548, 2011.

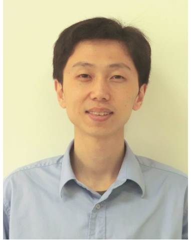
[25] M. Wing, D. Solmie, and L. Kellogg, "Comparing digital range finders for forestry applications," *Journal of forestry*, vol. 102, no. 4, pp. 16–20, 2004.

[26] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 631–640, 2010.

[27] Z. Zhong and T. He, "Achieving range-free localization beyond connectivity," in *Proceedings of ACM SenSys*. ACM, 2009, pp. 281–294.

Dr. Jizhong zhao is a Professor of Computer Science and Technology Department, Xi'an Jiaotong University. He received his BS degree and MS degree in Mathematic Department from Xi'an Jiaotong University. He received a Ph.D. degree in Computer Science, focus on Distributed System, from Xi'an Jiaotong University in 2001. His research interests include computer software, pervasive computing, distributed systems, network security. He is a member of IEEE computer society, and a member of ACM.

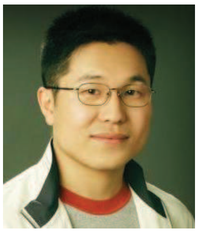
Wei Xi received his BS degree in School of Computer Science and Technology from Xidian University in 2006. He is currently a master-doctoral program graduate student at Xi'an Jiaotong University, and has been qualified as a PhD candidate in Department of Computer Science and Technology. His main research interests include wireless ad-hoc and sensor networks, and pervasive computing. He is a member of the IEEE and ACM.



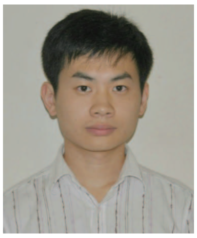
Dr. Yuan He received his BE degree in University of Science and Technology of China, his ME degree in Institute of Software, Chinese Academy of Sciences, and his PhD degree in Hong Kong University of Science and Technology. He is a member of Tsinghua National Lab for Information Science and Technology. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a member of the IEEE and ACM.

Dr. Yunhao Liu (SM06) received his BS degree in Automation Department from Tsinghua University, China, in 1995, and an MS and a Ph.D. degree in Computer Science and Engineering at Michigan State University in 2003 and 2004, respectively. He is a Professor at School of Software and Tsinghua National Lab for Information Science and Technology, Tsinghua University. He is also a faculty member at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology.

Dr. Xiang-Yang Li (M'99, SM'08) has been an Associate Professor (since 2006) and Assistant Professor (from 2000 to 2006) of Computer Science at the Illinois Institute of Technology. He received MS (1999) and PhD (2000) degree at Department of Computer Science from University of Illinois at Urbana-Champaign. He received the Bachelor degree at Department of Computer Science and Bachelor degree at Department of Business Management from Tsinghua University, China, both in 1995.



Lufeng Mo received his BE degree in Xi'an Jiaotong University, his ME degree in Pecking University. Mo Lufeng is currently working towards the Ph.D degree in the Department of Computer Science and Technology of Xi'an Jiaotong University. His research interests include wireless sensor networks and pervasive computing.



Dr. Zheng Yang (S'06) received his BS degree at Tsinghua University, China, in 2006 and a Ph.D degree in computer science at Hong Kong University of Science and Technology. He is currently a postdoc at the Tsinghua National Laboratory for Information Science and Technology and School of Software, Tsinghua University. His main research interests include wireless ad-hoc and sensor networks, and pervasive computing. He is a member of the IEEE and ACM.