Locating Sensors in the Wild: Pursuit of Ranging Quality

[1,3]Wei Xi, [2,3]Yuan He, [2,3]Yunhao Liu, [1]Jizhong Zhao
[1,4]Lufeng Mo, [2,3]Zheng Yang, [3]Jiliang Wang, [2]Xiangyang Li
[1]Xi'an Jiaotong University
[2]TNLIST, School of Software, Tsinghua University
[3]Hong Kong University of Science and Technology
[4]Zhejiang Agriculture and Forestry University

Abstract

Localization is a fundamental issue of wireless sensor networks that has been extensively studied in the literature. The real-world experience from GreenOrbs, a sensor network system in the 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. The central idea is that ranging quality is the key that determines the overall localization accuracy. In its unremitting pursuit of better ranging quality, CDL incorporates virtual-hop localization, local filtration, and ranging-quality aware calibration. We have implemented CDL and evaluated it by extensive experiments and simulations. The results demonstrate that CDL outperforms current state-of-art approaches with better accuracy, efficiency and consistent performance.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Localization, Wireless Sensor Network, Ranging Quality

1 Introduction

Localization is a crucial and critical task for 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 [3], under the ground [10], or in forests with dense canopies [13]. Range-based approaches measure the Euclidean distances among the nodes with various ranging techniques [19, 15, 20]. They are either expensive with respect to hardware cost, or susceptible to environmental dynamics [23]. Range-free approaches only rely on network connectivity measurements. However, localization results are typically imprecise and easily affected by node density. Mobile-assisted approaches have been proposed [3, 5, 24], but their use is constrained in certain environments.

This work is motivated by the need for accurate location information in GreenOrbs [13, 1], a large-scale forest based sensor network system. 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. Real-world experiences of GreenOrbs reveal that localization in the wild remains very challenging, in spite of the substantive efforts present in the literature. Non-uniform deployment of sensor nodes could affect the effectiveness of range-free localization. Furthermore, for range-based localization, the received signal strength indicators (RSSI) used for ranging 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 quality determines the overall localization accuracy. Bearing this in mind, recent proposals are focused more on error control and management. Some of those proposals enhance the localization accuracy by deliberately reducing the contribution of error-prone nodes to the localization process [12]. Other schemes improve localization by identifying large ranging errors and outliers, relying on topological or geometric properties of a network [25, 8].

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. The two aspects are equally important with respect to the impact on the localization results. Most of the proposals, however, address only one aspect, thus failing to achieve the desired localization accuracy.

To address the above 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, and keeps pursuing better ranging quality throughout the localization process. The contribu-

SenSys'10, November 3–5, 2010, Zurich, Switzerland.
Copyright 2010 ACM 978-1-4503-0344-6/10/11 ...$10.00

tions of this work are summarized as follows.

1. We propose a range-free scheme called virtual-hop localization, which carefully examines the local connectivity to mitigate the non-uniform node distribution problem. Using virtual-hop, the initial estimated node locations are more accurate than those output by other range-free schemes.

2. For better ranging quality, we devise two local filtration techniques, namely neighborhood hop-count matching and neighborhood sequence matching. The filtered good nodes provide high location accuracy.

3. Using the good nodes to calibrate the bad ones, we employ the technique of weighted robust estimation to emphasize the contributions of the best range measurements, eliminate the interfering outliers, and suppress the impact of ranges in-between.

4. We implement CDL with GreenOrbs and evaluate it with extensive experiments and simulations. The results demonstrate that CDL outperforms existing approaches with high accuracy, efficiency, and consistent performance.

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 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 [6], and DV-HOP [16], 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 [7], TDOA [19, 17], and AOA [3, 15]. 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 [22]. 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 [12].

More recent proposals mainly focus on the issue of error control and management [14, 11]. J. Liu *et al.* [12] 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.* [9] propose to assign different weights to range measurements with different nodes and adopt a robust statistical technique to tolerate outliers of range measurements. The noisy and outlier range measurement can be sifted by utilizing the topological properties of a network [8].

A range-free approach beyond connectivity is proposed in [25]. 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. According to node proximity, each node maintains a neighbor sequence. 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 (Section 4.1). Utilizing the information of estimated node locations, RSSI readings, and network connectivity, CDL filters good nodes from bad ones with two techniques (Section 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 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.

GreenOrbs currently collects four types of sensor data: temperature, humidity, illumination and carbon dioxide concentration. The collected data can be utilized to support a wide variety of applications, such as fire risk evaluation, canopy closure estimates [13], search and rescue in the wild, and the like. Specifically, it is always desired for GreenOrbs to determine potential fire risk areas, to monitor the canopy closure in geographically dispersed areas, to map the collected environmental factors to concrete vegetation, or to track moving objects (animals or human beings) in the forest. To meet these application requirements, location information of any given sensor node becomes crucial. Thus localization is a critical task for GreenOrbs.

To date, two GreenOrbs WSN systems have already been deployed; one in a campus woodland and the other in Tianmu Mountain, Zhejiang, China. This work is carried out mainly based on the first system, which currently includes 330 nodes in a deployment area of about $40,000m^2$. GreenOrbs nodes are deployed according to the following rule: the majority of nodes should be deployed where environmental information is required by forestry applications. The rest are then deployed to improve network connectivity. During the operational lifetime of the system, some nodes might need to change their locations so as to keep the network well connected or to obtain better sensing output from the monitored

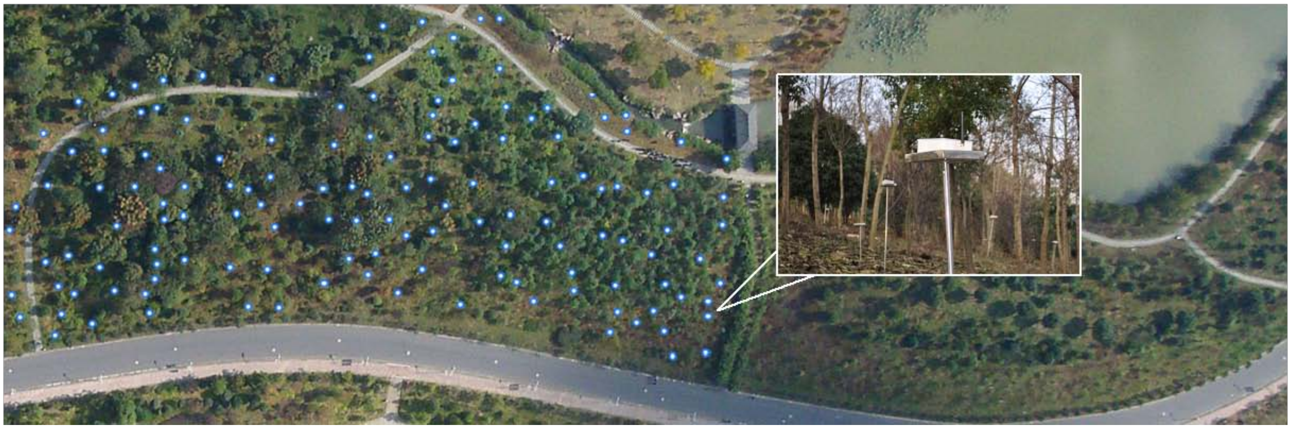


Figure 1. GreenOrbs deployment in the campus woodland

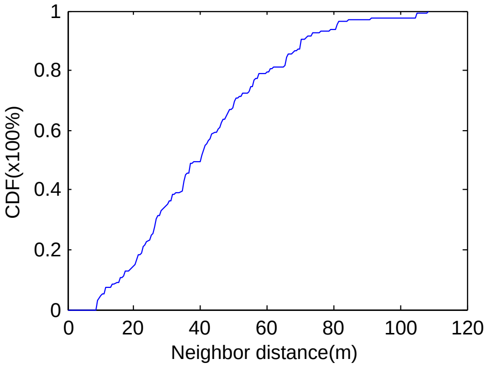


Figure 2. Cumulative distribution of node distances

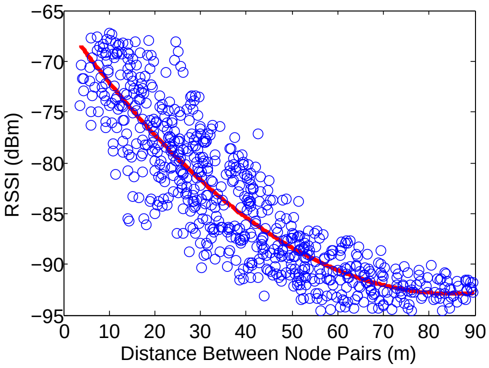


Figure 3. RSSI of different node pairs

area.

The ground-truth coordinates of the nodes are measured using an EDM (Electronic Distance Measuring Device) [4]. This requires careful mounting of the EDM on the forest floor and cooperation between two surveyors. The measurement process is hence very laborious and time-consuming. So far we have succeeded in measuring the coordinates of 100 nodes, as shown in Figure 1. Thus the observations and experiments in this paper are mainly conducted using those 100 nodes. The other 230 nodes, although deployed in an adjacent area, are not shown in the picture.

3.2 Observations

As we can see from Figure 1, sensor nodes are deployed on the forest floor with most of them under dense tree cover, where GPS usually does not work [7]. Even in areas with less dense tree cover, our experience shows the errors produced by a portable GPS device (compared to an EDM) are often as large as 15m. Thus locating nodes basically comes down to in-network localization, namely range-based or range-free approaches. This subsection presents real-world observations on GreenOrbs. The observations illustrate that a single approach, whether it is range-based or range-free, has apparent 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, so as to provide comprehensive and fine-grained information of the monitored area. Such a rule leads to non-uniform deployment of sensor nodes, as we can see from Figure 1. Specifically, some nodes have more than 20 neighbors while some nodes have less than 5.

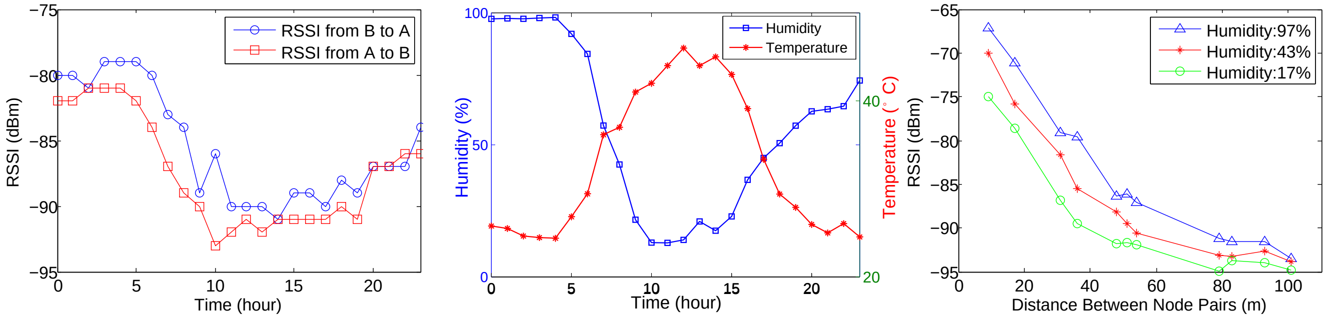
The distances between pairs of neighboring nodes also differ a lot, as shown in Figure 2. The shortest distance is 5m and the longest is around 108m.

Range-free localization (for example DV-Hop) in a non-uniform deployment often incurs large errors. Two nodes are located to the same position if they have identical hop-counts to the landmarks. Nevertheless, in reality they might be some distance from each other.

3.2.2 Irregularity of RSSI

Besides the non-uniform deployment problem, complex terrain and obstacles (for example shrubs and tree trunks) also affect signal propagation in the forest. The resulting RSSI among the nodes is very irregular. Figure 3 plots the RSSI between many pairs of nodes in GreenOrbs at a certain time. It further includes a curve, which shows the mapping between RSSI and the distance based on the log normal shadowing model.

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

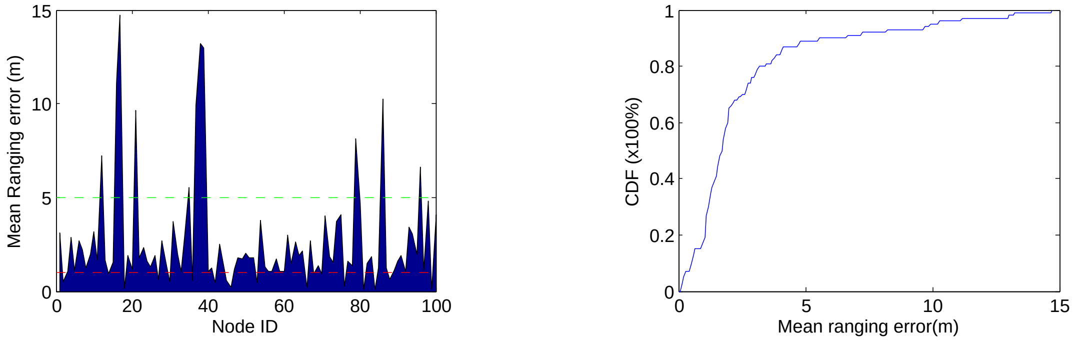


(a) RSSI between nodes A and B

(b) The corresponding environmental dynamics

(c) Mean RSSI vs. varying humidity

Figure 4. Asymmetry and dynamics of RSSI



(a)

(b)

Figure 5. Errors of range measurements on the nodes

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 $X{\sim}N(0,\sigma_{X^2})$. For all the experiments in this work, we set the parameter values as $\eta = 3.3$, $X_\sigma = 6$, according to the empirical results reported in [18].

We can see that the real distances between node pairs differ greatly from the model-based estimations. The mapping of the RSSI distance is actually very uncertain. Yet RSSI still offers useful information. In most cases, a stronger RSSI corresponds to a shorter distance, as is also observed in [5, 25].

3.2.3 *Asymmetry and Dynamics of RSSI*

Figure 4 shows the pairwise RSSI between two nodes in GreenOrbs, along with the temperature and humidity 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 probably have unequal RSSI. Moreover, RSSI is very susceptible to environmental factors, such as humidity and temperature. At different times of a day, the RSSI over a link is highly variable.

3.2.4 *Errors of RSSI-based range measurements*

We randomly select a time to collect the RSSI readings from all 100 nodes. The RSSI are then used for range measurements based on Equation (1). By comparing the converted distances with ground-truth, Figure 5 plots the ranging error of each node. The ranging error is the mean error of range measurements between a node and each of its neighbors. Generally, if a node has line-of-sight connections with all its neighbors, the mean error of its range measurement is small. If a node is mounted on the trunk of a large tree, lies in a pit, or has a faulty antenna, the mean error is likely to be large. To indiscriminately use the log normal shadowing model of RSSI to locate sensors in the wild may introduce considerable errors.

Interestingly, the fact is that only 9% of the nodes have large ranging errors ($>$ 5m) and only 18% of the nodes have small ranging errors ($<$ 1m). The errors of the rest of the nodes (73%) are between 1m and 5m. Such errors cannot be easily detected but they seriously degrade the overall localization accuracy.

Here is a brief summary of observations on GreenOrbs. First, the sensor nodes are deployed with diverse densities in different regions, causing the non-uniform distribution problem. Second, RSSI over the wireless links is very unstable and sensitive to various environmental factors. The uncertainty of RSSI is hard to model in practice, therefore; as a result RSSI-based range measurements exhibit quite diverse errors. Third, only a small portion of range measurements are accurate. To make matters even worse, typically only large ranging errors can be detected or tolerated by the existing approaches.

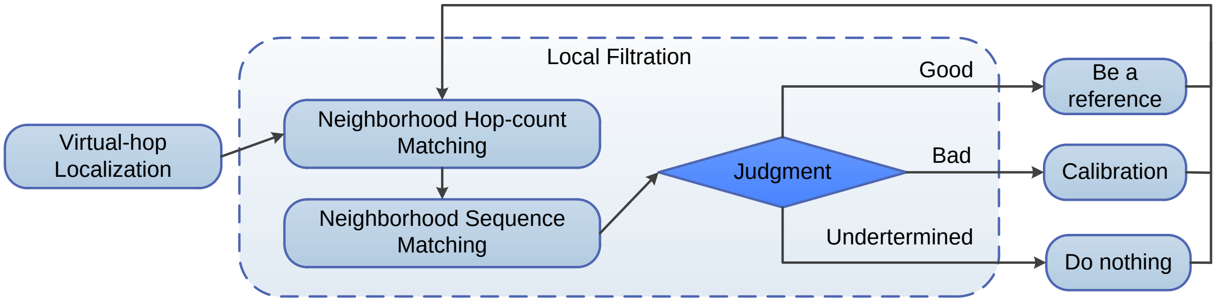


Figure 6. The work flow of CDL

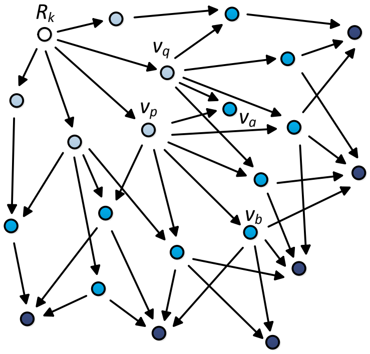


Figure 7. Nodes with the same hop counts have different distances to the landmark

4 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*. Figure 6 illustrates the CDL workflow.

Virtual-hop localization initially estimates node locations using a range-free method. In order to approximate the distances from the nodes to the landmarks, it counts 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.

Table 1. Variable Definitions

Symbol	Definition
$\Re_j$	a set of all neighboring nodes of node v_j
P_j	a set of previous-hop neighbor of node v_j
N_j	a set of next-hop neighbor of node v_j
h_{ij}	the hop count from node v_i to node v_j
ζ_j	$Min\{\|P_i\| \mid v_i \text{ is in } N_j\}$
φ_j	$Min\{\|N_i\| \mid v_i \text{ is in } P_j\}$

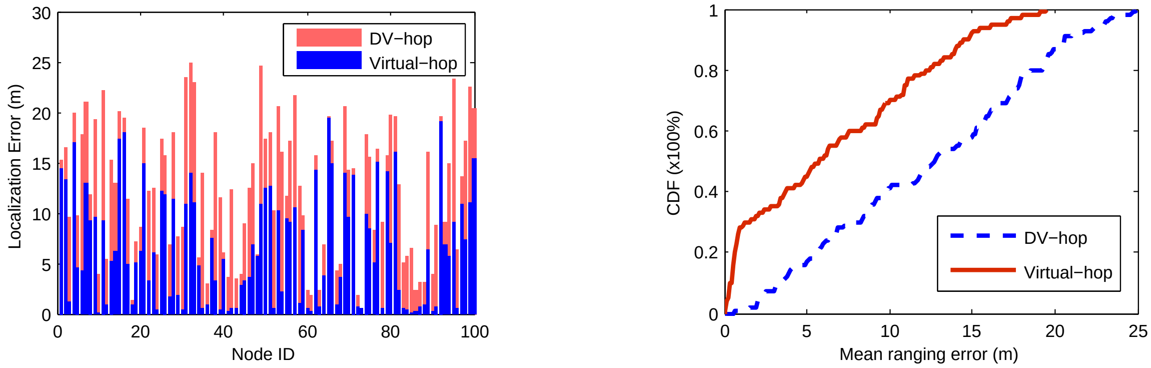
4.1 Virtual-Hop Localization

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 and improves the localization accuracy in such contexts. 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 Virtual-hop

For ease of presentation, Table 1 lists several symbols and notations we use in this paper. To illustrate the virtual-hop algorithm more clearly, we give an example based on Figure 7. It illustrates an interesting fact. For two nodes that have equal hop counts to the landmark, the node nearer to the landmark is likely to have more previous-hop neighbors and less next-hop neighbors. Let us compare nodes v_a and v_b in the figure. They are both two hops from the landmark, but v_a is apparently closer to the landmark than v_b. v_a has more previous-hop neighbors and less next-hop neighbors. This can be represented as following: $|P_a| = 2$ and $|N_a| = 0$, while $|P_b| = 1$ and $|N_b| = 3$.

Based on above observations, we propose the design of virtual-hop localization. Each node maintains two types of hop counts. One of them is the real hop count, based on which a node determines its previous-hop and next-hop neighbors. The other is the virtual-hop-count, which is calculated hop by hop. VH_{jk} denotes the virtual hop count from landmark R_k to node v_j. It is the sum of two parts. The first part is the average virtual-hop-count of VH_{jk}'s previous-hop neighbors. The second part is the last virtual-hop-count, that is, the incremental virtual-hop-count from VH_{jk}'s previous-



(a) (b)

Figure 8. Comparison of DV-hop and Virtual-hop

hop neighbors to VH_{jk}, denoted by VH_{jk}. We have

$$VH_{jk} = \frac{1}{|P_j|}\sum_{i=1}^{|P_j|} VH_{ik} + LH_{jk} \tag{2}$$

where

$$LH_{jk} = \begin{cases} \dfrac{|N_j|}{|N_j| + \zeta_j - 1}, |N_j| > 0 \\ \dfrac{|P_j|}{|P_j| + \varphi_j - 1}, |N_j| = 0 \end{cases}$$

By using the numbers of previous-hop neighbors and next-hop neighbors of relevant nodes as input parameters in Equation (2), VH_{jk} is quantified as the relative length of the last virtual-hop of v_j. Here we use nodes v_a and v_b as examples to illustrate the improvement to the virtual-hop in figure 7. Node va has two previous-hop neighbors v_p and v_q, while v_b has only one previous-hop neighbor v_p.We can calcite the virtual-hop of v_a and v_b as follows.

$VH_{pk} = 6/(6+1-1) = 1, VH_{qk} = 5/(5+1-1) = 1.$

$LH_{ak} = 2/(2+5-1) = 0.33.$

$VH_{ak} = (VH_{pk} + VH_{qk})/2 + LH_{ak} = 1.33.$

Similarly we get $VH_{bk} = 1 + 0.8 = 1.8.$

After that, the per-virtual-hop distance regarding landmark R_k is calculated by

$$\begin{aligned} \tilde{d_k} &= \frac{\sum \rho_{tk}}{\sum VH_{tk}} \; where\, R_t \, is\, a\, landmark,\, t \neq k. \\ &= \frac{\sum \sqrt{(X_t - X_k)^2 + (Y_t - Y_k)^2}}{\sum VH_{tk}} \end{aligned} \tag{3}$$

where ρ_{tk} is the Euclidean distance between landmarks R_t and R_k. Each node computes its distance to the landmarks by

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

where ρ_{jk} is the estimated distance from v_j to R_k. After calculating the distances to the landmarks, each node computes its coordinates based on trilateration using Least Square Estimation (LSE), which is similar to DV-hop.

4.1.2 *Localization Accuracy of Virtual-hop*

We now look at our simulation results to compare the accuracy of virtual-hop localization with DV-hop. The setup includes 100 nodes and 4 landmarks. The simulation results are shown in Figure 8. We can see virtual-hop outperforms DV-hop remarkably. The performance gain of using virtual-hop varies a lot among different nodes. Compared with DV-hop, the localization errors are reduced by $10\% \sim 99\%$.

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

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.

4.2.1 *Infeasibility 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 before calibration. We call this straightforward model-based calibration *indiscriminate calibration*. Using such calibration, every node’s location is adjusted directly based on the distances to its neighbors converted from RSSI, using the log-normal shadowing model.

Figure 9 compares the localization errors of nodes before

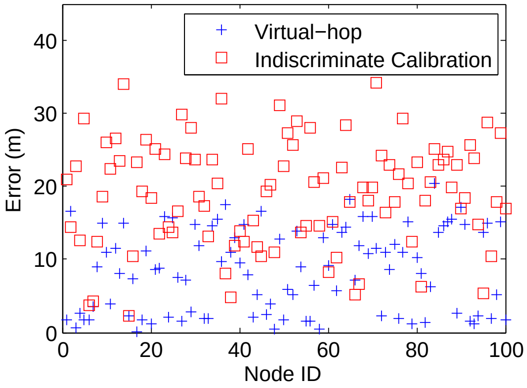


Figure 9. The incorrectness of Indiscriminate Calibration

and after indiscriminate calibration. Surprisingly, we find the output of *indiscriminate calibration* 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 node 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, $d_{ij} = d'_{ij}$. 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 the difference $|d_{ij} - d'_{ij}|$ corresponding to every neighbor v_j, we can measure the *Aggregated Degree* of Mismatches (ADM) of v_j.

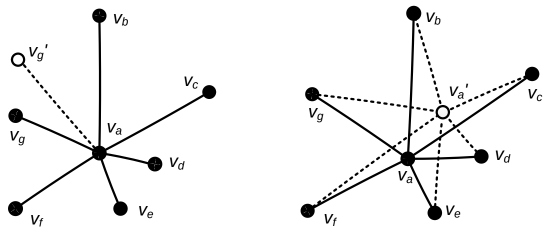
ADM actually reflects the error of a node's estimated location. For example in Figure 10 (a), node v_a is a good node with six neighbors. Among them only node v_g is a bad node. Let v'_g denote its estimated location. Clearly the ADM of node v_a is mainly caused by v'_g. In Figure 10 (b), node v_a is a bad node with six good neighbors. The link to every neighbor contributes to the ADM of node 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.

The above method to quantify ADM, however, relies on the log-normal shadowing model to convert RSSI into a distance. As we have observed, such conversion is error-prone, which leads to totally incorrect filtration.

4.2.2 Neighborhood Hop-Count Matching

Every node takes neighborhood hop-count matching as the first step to identify whether it is a bad node. This mainly utilizes 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 node v_i as an example to illustrate the matching procedure.

First, every node exchanges its estimated coordinates with its 2-hop neighborhood.



(a) A good node with one bad neighbors

(b) A bad node with six good neighbors

Figure 10. ADM reflects the localization error of a node

Second, when v_i receives the estimated coordinates of v_j, it 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 $h'_{ij} = \lceil d'_{ij}/\tilde{d}_k \rceil$, where $\tilde{d}_k$ is the per-hop distance obtained from the nearest landmark R_k during virtual-hop localization.

Fourth, v_i computes its ratio of matched hop-counts within its 2-hop neighborhood v_j as follows:

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

$$r_i = \frac{1}{n} \sum_{j=1}^{n} H_{ij} \tag{6}$$

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

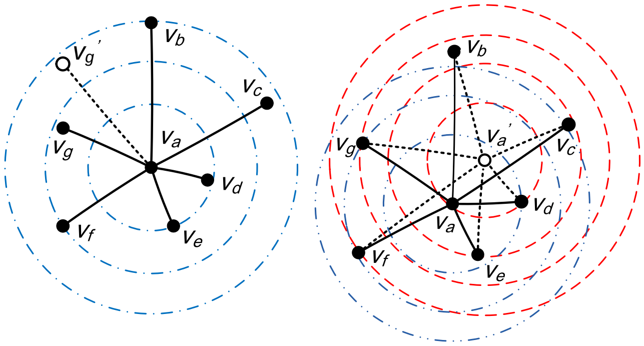
where h_{ij} denotes the hop count from v_i to v_j and n is the number of its 2-hop neighbors of v_i. $\overline{r}_i$ denotes the mean matched ratio in the neighborhood of v_i. If $r_i < \overline{r}_i$, v_i regards itself as a bad node, which has an apparent error in its estimated coordinates. Otherwise,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

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 node distance increases, as can be observed from the RSSI readings in Figure 3. Based on this observation, we propose a filtration scheme called *neighborhood sequence matching*. The filtration on a node v_a runs as illustrated in Figure 11.

First, node 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.



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) A good node with one bad neighbor

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) A bad node with six good neighbors

Figure 11. Neighborhood sequence matching

Second, according to the estimated coordinates, node 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 theory, 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 Figure 10 to illustrate the above idea. As shown in Figure 11 (a), v_a is a good node while v_g is the only bad neighbor whose location is mistaken as v_g'. There is not a significant mismatch between S_a and S_a' in this case. Comparatively in Figure 11 (b), when v_a is a bad node with six good neighbors, there appears to be significant mismatch between S_a and S_a'.

Now that the difference between S_a and S_a' is a measure of the location error of v_a, the next step is to quantify their distance. The cosine distance is a measure of similarity between two vectors by finding the cosine of the angle between them. One may use the cosine distance to measure the similarity between sequences S_a and S_a'. However, when the cosine distance is used alone, it cannot sufficiently distinguish good nodes from bad nodes. Specifically, 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, so we define the matching degree M_i between the *RSSI sequence* and *distance sequence* as follows.

$$M_i = \delta_i \cdot CosDist_i \tag{8}$$

where n is the number of v_a's neighbors, δ_i is the proportion of the length of the longest common subsequence (LCS) between S_i and S_i' and n, *CosDist* is cosine distance of two sequences, $a_1, a_2, ..., a_n$ are the sequence numbers in

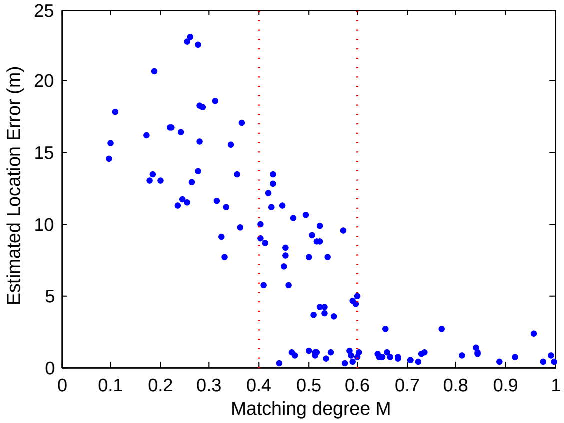


Figure 12. Filtration result of virtual-hop localization

S_a while $a_1', a_2', ..., a_n'$ are the sequence numbers in S_a'. Since $a_1, a_2, ..., a_n = a_1', a_2', ..., a_n' = 1, 2, ..., n$, we have:

$$\begin{aligned} CosDist &= \frac{a_1a_1' + a_2a_2' + ... + a_na_n'}{\sqrt{a_1^2 + a_2^2 + ... + a_n^2}\sqrt{a_1'^2 + a_2'^2 + ... + a_n'^2}} \\ &= \frac{a_1a_1' + a_2a_2' + ... + a_na_n'}{1^2 + 2^2 + ... + n^2} \end{aligned} \tag{9}$$

Clearly M_i is a better metric to distinguish good nodes from bad nodes. In Equation (9), a_1, a_2, ..., a_n 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 a_1, a_2, ..., a_n, a_1', a_2', ..., a_n', and 1, 2, ..., n are three equal sets.

Generally, when RSSI readings have small errors, *RSSI sequence* of a good node still well matches its *distance sequence*. Even when a small portion of RSSI readings have relatively large errors, the matching degree cannot be influenced much. Thus the scheme of *neighborhood sequence matching* is error-tolerant to interference from background noise.

We use the same trace as that in Figure 8 to calculate the matching degree of all the nodes after initial localization. The results are plotted in Figure 12. 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. It is by far too hard to decide whether they are good or bad. Their matching degree is between 0.4 and 0.6, Thus we tentatively set them as undetermined nodes. $\tau_l = 0.4$ and $\tau_u = 0.6$ are two empirical parameters, called the lower matching threshold and the upper matching threshold. 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, respectively.

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

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

Given the range measurements between bad node v_i and its good neighbors, the estimation of v_i's location, denoted by f^*, usually works by minimizing an objective function 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 (11).

When least square estimator (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) [9] outperforms LSE by assigning different weights to the range measurements with different neighbors.

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

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 situations where the majority of range measurements are accurate. Otherwise it becomes inefficient, as is shown by the experiments in Section 5.

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

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

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

where θ is a pre-configured parameter. 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 (calculated using Equations (14) and (15)) is good. Otherwise, the weight of the good node will be relatively low. The objection function of RQAC is defined as follows.

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

Note that $|l_{ij} - d_{ij}|$ is a measure of the ranging error. ω_j and $|l_{ij} - d_{ij}|$ thus jointly denote the ranging quality from v_j to v_i.

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 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 $|l_{ij} - d_{ij}|$. Note that $g(i, j)$ is continuous at $|l_{ij} - d_{ij}| = \varepsilon$. 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.

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 the effectiveness of 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.

5 EVALUATION

We have implemented CDL with GreenOrbs. The performance of CDL is evaluated through large-scale real experiments as well as simulations. For comparison, we have also implemented three existing localization approaches, namely DV-hop [16], MDS-MAP(C,R) [21], and SISR [9].

Specifically in MDS-MAP(C,R), the 1-dimensional MDS is applied first. Then four corner nodes are used as landmarks for the scaling and rotation of the map in 2-dimensional MDS. MDS-MAP(C,R) avoids the problem that multiple nodes are localized to the same location. Its localization performance is usually better than DV-hop. However, bad links cause big errors in the range measurements. This problem

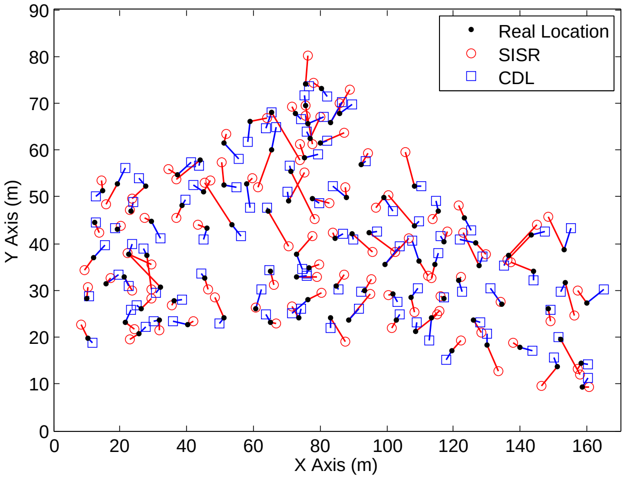


Figure 13. Localization results of SISR and CDL

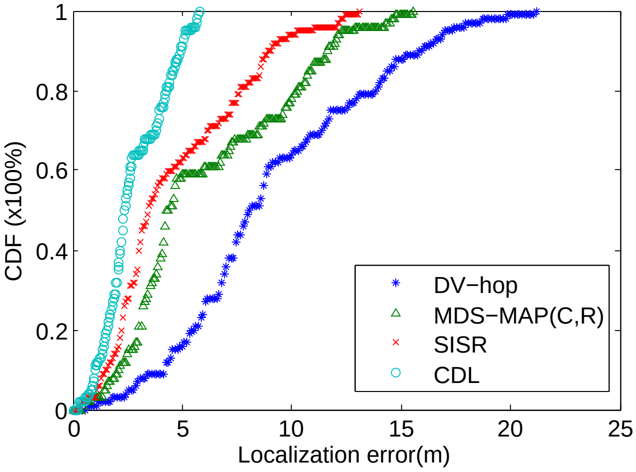


Figure 14. Overall localization results

remains unresolved in MDS-MAP(C,R). The estimated coordinates of some nodes still often deviate significantly from the real ones.

Corresponding to the deployment map in Figure 1, Figure 13 plots the 100 GreenOrbs nodes in a rectangular region. Four nodes positioned near the border of the deployment area are selected as landmarks. We have collected the localization results of all the four approaches.

5.1 Experiments

5.1.1 Comparison among Approaches

Figure 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 Figure 13.

Figure 13 shows that for almost all the nodes, CDL achieves higher localization accuracy than SISR. A detailed explanation of the results can be found in Figures 14 and 15.

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 Figure 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.

5.1.2 Efficiency of Iteration

Note that CDL and SISR both propose iterative localization processes. Other than comparing the overall performance, we conduct experiments to evaluate the efficiency.

As shown in Figure 15, the mean localization errors of CDL and SISR both decrease as the iterations go on. Their performances converge after 6 $\sim$ 8 rounds of iterations. The localization accuracy of CDL is always better than that of SISR.

We want to emphasize that, in setting an objective of localization accuracy, the required number of iterations actually determines the communication and computational cost of a localization approach. By examining the results in Figure 15, we are pleased to see CDL achieves very satisfactory localization results even after only two rounds of iterations. The performance gain of a few more rounds of iterations is also more apparent with CDL than with SISR. The accuracy of SISR is constantly improved until the 4th round of iteration.

CDL outperforms SISR mainly because of the following reasons.

First, the initial round of CDL iterations starts with the output of virtual-hop localization, while SISR starts from totally undetermined locations. With only straightforward location adjustment, SISR cannot converge sufficiently quickly. As a result, the localization error of SISR in the first few rounds of iterations is especially large.

Second, filtration and calibration in CDL explicitly filter the bad nodes and identify the good nodes and good ranging quality. Hence, CDL may maximize the contribution of those good nodes in a calibration and clearly eliminate the negative impact of bad nodes. Such a capacity is especially preferred during the first few rounds of iterations, when the number of good nodes is relatively limited.

On the other hand, SISR conducts calibration without explicit differentiation on the quality of ranging. As we observe with GreenOrbs, there are many mild ranging errors as shown in Figure 5. SISR is effective only when there is a relatively small portion of sizable ranging errors. When

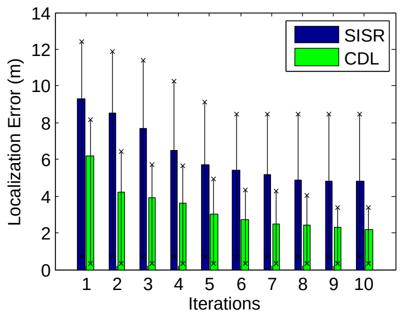
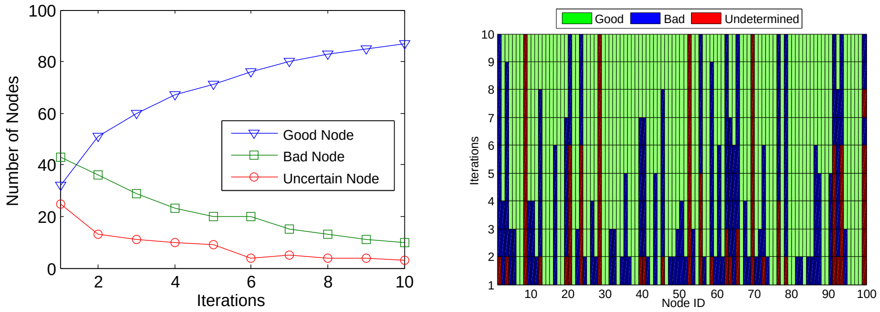


Figure 15. Comparison of SISR and CDL with iterations



(a) Node type distribution

(b) Evolution of node types

Figure 16. Node type distribution

sizable ranging errors widely exist with most of the nodes, SISR cannot accurately distinguish between bad nodes and good ones. The consequence is inefficiency in the starting stage of iterations.

We may notice that after some rounds of iterations, the localization errors of CDL reach a relatively stable value and no longer goes down. The errors remaining in the localization results are mainly incurred by a small portion of bad nodes which cannot be better located. Those bad nodes do not have a sufficient number of good neighbors or are likely to have potential ranging errors to almost all their good neighbors. When usable information from the good neighbors is used up, the localization errors converge. This is a common fact with CDL and SISR. Increasing the transmission power might be a simple but effective method to improve such a situation. Since this is not the key issue to be studied in this paper, we leave it for future work.

5.1.3 *Evolution of Node Types*

In this subsection, we observe the evolution of node types, namely good, bad, or undetermined, along with the iterative filtration and calibration of CDL.

Figure 16 (a) shows the trends in the changing numbers of good, bad, and undetermined nodes along with 10 iterations. First of all, CDL makes deterministic judgments on node types (good or bad) for almost all the nodes after 10 rounds of iterations. The effect of calibration is also satisfactory. We can see the number of good nodes quickly increases as iterations go on.

In order to avoid false positive and negative filtration, we conservatively leave some nodes with medium values of matching degrees. Hence there remains a considerable amount of undetermined nodes in each of the first iterations. As the filtration and calibration continue, more and more bad nodes are successfully calibrated by the good ones. The uncertainties of node location decrease, so the undetermined nodes gradually obtain the opportunity to be identified and calibrated.

Figure 16 (b) shows the detailed evolutional process of all the 100 nodes in 10 iterations. Specifically, 31% nodes are correctly identified to be good from the very beginning. 44% nodes are initially identified as bad, and then calibrated to be good. 25% of them are initially undetermined, then identified as bad nodes, and finally calibrated to be good. The rest of the nodes, which count for only 4%, are left undetermined at the end of 10 rounds of iterations.

5.1.4 *The Impact of Environmental Factors*

Our observations in Section 3 show that the dynamics of environmental factors have significant impact on RSSI among the nodes. RSSI is concerned with the range-based calibration. It also affects the transmission range of nodes. Therefore, the dynamics of RSSI are also concerned with the accuracy of virtual-hop localization. Thus, it would be interesting to investigate how the localization approaches perform under different environmental conditions.

Figure 17 plots the localization results of the four approaches under two representative conditions in GreenOrbs. Since GreenOrbs collects humidity data, we are able to monitor the humidity over time while carrying out localization experiments.

Actually, RSSI is affected by both temperature and humidity. Note that in Figure 4 (b), temperature and humidity presents completely inverse trends of variation. We believe the humidity data alone is sufficient for us to evaluate the impact of both humidity and temperature.

As we can see from Figure 17, all four approaches achieve better performance with higher humidity. Recall the experience reported in [13], RSSI increases with increasing humidity. A larger RSSI results in a longer transmission range, so that a node has more neighboring nodes, and in turn has better chance of being calibrated. The problem with non-uniform deployments is also mitigated, because a node obtains more information of locations and connections from its neighborhood.

This group of experiments implies that environmental factors do matter in localization. One may expect to achieve higher localization accuracy in weather conditions when the wireless signals are relatively stable and strong. When we make judgments on the quality of a localization approach, the environmental conditions of the localization being executed should definitely be considered.

5.2 Simulation

Besides the above experiments based on the implementation of CDL in GreenOrbs, we have carried out extensive simulations to evaluate the performance of CDL. We examine the location accuracy of CDL by tuning a series of parameters such as node density, the number of landmarks, and the

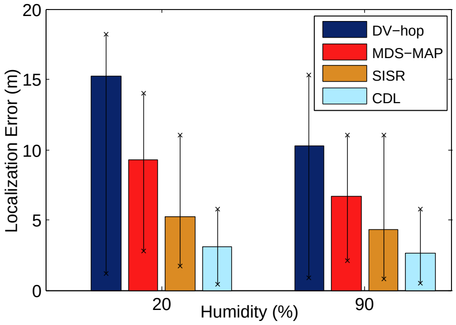


Figure 17. Impact of humidity

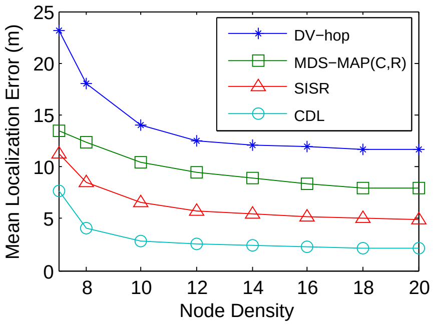


Figure 18. Impact of node density

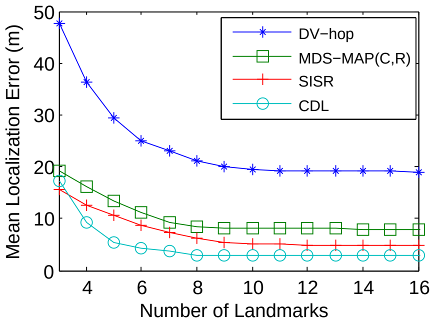


Figure 19. Impact landmarks count

relative ranging errors. The simulation results of DV-hop, MDS-MAP(C,R), and SISR are also presented for a more comprehensive comparison.

In the simulations, nodes are randomly deployed in a 500m*500m square region. We set the transmission range of nodes to be 30m.

5.2.1 *Node Density*

The node density (ND) is the average number of one hop neighbors of a node. It is often an important factor in localization. In this group of simulations, we keep the number of landmarks constant in the network and tune the node density by changing the number of nodes. We set the number of landmarks equal to 6 and the percentage of bad links to 25%. To simulate the range measurements on the bad links, we let the relative ranging errors on the links conform to a Gaussian distribution, denoted by $N(0.5, 0.1)$.

Figure 18 compares the converged value of mean localization errors using the four approaches. For DV-hop and MDS-MAP(C,R), since they are not iterative methods, we just present their localization results from one time of execution. The curves in Figure 18 show that a higher node density results in higher location accuracy for all the four approaches.

Specifically, DV-hop performs very poor when the node density is low. This is because the non-uniform deployment problem appears to be more serious with sparse deployments. When the node density increases, the non-uniform deployment problem is mitigated and the localization accuracy is thus improved.

For MDS-MAP(C,R), a high node density makes the total length of a shortest path between two nodes correspond well to their Euclidean distance. With a higher node density, it is also easier for SISR to obtain enough good range measurements for localization.

As for CDL, it is least susceptible to low node density among the four approaches, because virtual-hop localization already addresses the non-uniform deployment problem to a certain extent. Another advantage of CDL is that even with a limited number of good nodes, it still can make efficient filtration and calibration.

Nevertheless, node density does matters for CDL as well. As the node density goes up, the mean localization error quickly drops and then gradually converges. A higher node density usually results in better accuracy of virtual-hop localization. Moreover, as the node density increases, the chances for CDL to make more efficient filtration and calibration also get better.

5.2.2 *Impact of the Number of Landmarks*

Due to the complexity of deployments and location measurements of WSNs in the wild, there are usually a limited number of landmarks. Thus it is worth examining how CDL works with different numbers of landmarks. In our simulation, we tune the number of landmarks from the minimum required number (that is, 3) to a relatively large number, namely 16. The total number of nodes is 1000. The node density $ND = 12$ and the percentage of bad links is 25%. All four approaches perform well with such settings, according to the previous experimental results.

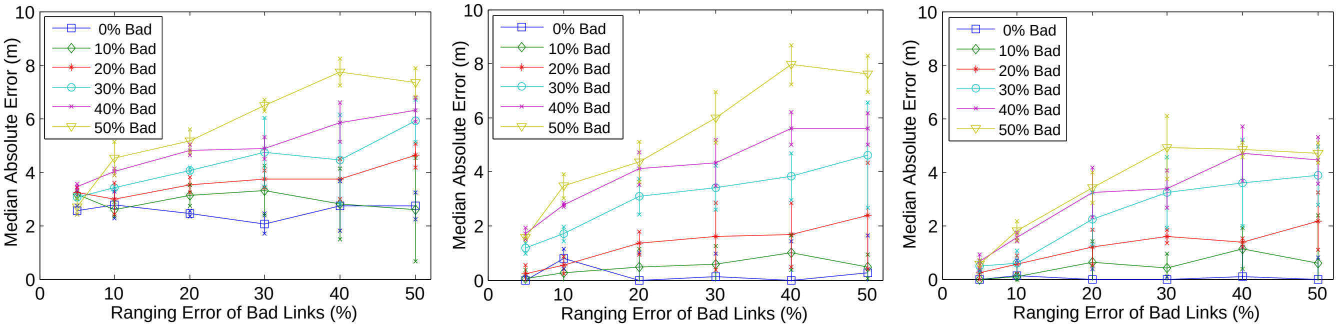
Figure 19 plots the mean localization errors of the four approaches as the number of landmarks is increased. We can see that MDS-MAP(C,R) and SISR are less sensitive to the number of landmarks than DV-hop and CDL. The main reason is that landmarks are not involved throughout the localization process with MDS-MAP(C,R) and SISR. For example, SISR first generates a relative location map of the network and then transforms it to absolute positions when sufficient landmarks are available.

On the other hand, DV-hop simply relies on counting hops to the landmarks to calculate the node locations. CDL is also a bit affected by the number of landmarks, because it includes virtual-hop localization as a component. Yet we notice that CDL performs slightly worse than SISR only when there are three landmarks. The localization accuracy of CDL is improved remarkably by adding one or two landmarks to the network. Overall, six landmarks are more than enough for CDL to achieve satisfactory performance.

5.2.3 *Ranging Error*

Considering the ubiquitous ranging errors and poor overall ranging quality 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, we conduct a group of simulations with 1000 nodes. Node density ND is set to 12 and the number of landmarks to eight.

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



(a) MDS-MAP(C,R) (b) SISR (c) CDL

Figure 20. Comparison of localization errors

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.

Figure 20 plots the mean localization errors of MDS-MAP (C,R), SISR, and CDL under different settings. We find MDS-MAP (C,R) is relatively insensitive to the changes of ranging errors, because it mainly relies on connectivity information instead of inter-node ranging. All MDS-MAP(C,R) results have localization errors of more than 2m, even when all links are good.

SISR generally performs better than MDS-MAP(C,R) and 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 perform robustly enough. The mean localization error is around 5m. This group of simulation shows the remarkable advantages of CDL in extremely complex environments.

5.2.4 *Overhead Analysis*

Though cost is not the first concern of localization, we analyze the communication cost 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 $O(m)$. Each node exchanges relevant information with its one-hop neighbors to estimate the 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 $O(k)$.

In local filtration, the major communication cost of a node is incurred by information exchange with its one-hop/two-hop neighbors. Thus the communication cost in this phase is $O(k^2)$.

n RQAC, all cost is incurred by local computation and thus ignorable, compared to the communication costs in the previous two phases.

6 CONCLUSION

Localization has received substantive attention in the research community over the past decade. Yet the state-of-the-art schemes remain immature indeed, especially when it comes to real-world WSNs in complex environments. This paper presents our experience of localization with GreenOrbs, a system in the forest with a number of factors that can interfere with localization, such as environmental dynamics, obstacles, channel noise, and signal irregularities.

Aiming at efficient and accurate localization, our design called CDL is engaged in a step-by-step process to pursue the best ranging quality. We have implemented CDL and carried out extensive experiments and simulations. The results demonstrate that CDL outperforms existing approaches with high accuracy, efficiency, and consistent performance in the wild.

We share our real-world experience, design, and evaluation with the community. Though this work cannot be generalized to every possible case, the community may well benefit from our understanding of the practical challenges of localization in the wild as well as promising directions. GreenOrbs will continue the pursuit of ranging quality in its future expansion and development.

7 Acknowledgements

The authors would like to thank the shepherd, Akos Ledeczi, for his constructive feedback and valuable input. Thanks also to anonymous reviewers for reading this paper and giving valuable comments.

This work is supported in part by China 863 Program under grant No.2009AA011903, NSFC under Grant No.60828003, NSFC/RGC Joint Research Scheme N-HKUST602/08, China 973 Program under Grants No.2011CB302705 and No.2010CB334707, NSF CNS-0832120, Program for Zhejiang Provincial Key Innovative Research Team, Program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

8 References

[1] `http://www.greenorbs.org`.

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

[3] H. Chang et al. Spinning beacons for precise indoor localization. In *Proceedings of ACM SenSys*, pages 127–140. ACM, 2008.

[4] 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*, 35(8):1455–1460, 2003.

[5] 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*, pages 93–102. IEEE, 2008.

[6] 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*, pages 81–95. ACM, 2003.

[7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. Global positioning System. Theory and Practice. 1993.

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

[9] 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*, pages 333–344. ACM, 2009.

[10] M. Li and Y. Liu. Underground structure monitoring with wireless sensor networks. In *Proceedings of ACM/IEEE IPSN*, pages 69–78. ACM, 2007.

[11] 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*, pages 91–98. IEEE, 2005.

[12] J. Liu, Y. Zhang, and F. Zhao. Robust distributed node localization with error management. In *Proceedings of ACM MobiHoc*, pages 250–261. ACM, 2006.

[13] 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*, pages 99–112. ACM, 2009.

[14] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of ACM SenSys*, pages 50–61. ACM, 2004.

[15] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proceedings of IEEE INFOCOM*, pages 1734–1743. IEEE, 2003.

[16] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22(1):267–280, 2003.

[17] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beep-Beep: a high accuracy acoustic ranging system using COTS mobile devices. In *Proceedings of ACM SenSys*, pages 59–72. ACM, 2007.

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

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

[20] S. Seidel and T. Rappaport. 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE transactions on Antennas and Propagation*, 40(2):207–217, 1992.

[21] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):961–974, 2004.

[22] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from mere connectivity. In *Proceedings of ACM MobiHoc*, pages 201–212. ACM, 2003.

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

[24] W. Xi, J. Zhao, X. Liu, X. Li, and Y. Qi. EUL: An efficient and universal localization method for wireless sensor network. In *Proceedings of IEEE ICDCS*, pages 433–440. IEEE Computer Society, 2009.

[25] Z. Zhong and T. He. Achieving range-free localization beyond connectivity. In *Proceedings of ACM SenSys*, pages 281–294. ACM, 2009.