



Distributed Push–pull Estimation for node localization in wireless sensor networks

Viet-Hung Dang, Viet-Duc Le, Young-Koo Lee, Sungyoung Lee*

Department of Computer Engineering, KyungHee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Republic of Korea

ARTICLE INFO

Article history:

Received 24 March 2010
Received in revised form
30 June 2010
Accepted 2 July 2010
Available online 1 August 2010

Keywords:

Distributed localization
Push–pull Estimation
Virtual force

ABSTRACT

A great deal of research achievements on localization in wireless sensor networks (WSNs) has been obtained in recent years. Nevertheless, its interesting challenges in terms of cost-reduction, accuracy improvement, scalability, and distributed ability design have led to the development of a new algorithm, the Push–pull Estimation (PPE). In this algorithm, the differences between measurements and current calculated distances are modeled into forces, dragging the nodes close to their actual positions. Based on very few known-location sensors or beacons, PPE can pervasively estimate the coordinates of many unknown-location sensors. Each unknown-location sensor, with given pair-wise distances, could independently estimate its own position through remarkably uncomplicated calculations. Characteristics of the algorithm are examined through analyses and simulations to demonstrate that it has advantages over those of previous works in dealing with the above challenges.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

Wireless sensor networks (WSNs) are sets of sensors spatially deployed mainly for applications of observing, tracking, and controlling. Observing includes collecting data like parameters of temperature, humidity, vibration, pressure, traffic, event counting, etc. Tracking uses the observed data to follow and sometimes to predict changing data where moving object tracking is an example. Meanwhile, controlling pertains to decisions in response to data provided by observing and tracking, like cutting power, sending instructions to control traffic, putting out fires, sending alarms, releasing missiles to detected enemy devices, and so on. Evidently, applications of WSNs are all related to the physical locations of sensed data.

The position of a sensor, or the prerequisite information, can be achieved easily and correctly with an integrated Global Positioning System (GPS) module. However, there is one problem with GPS making localization algorithms necessary in practical implementation. It is the expensive cost of GPS devices on sensors, causing the total cost considerably high when the number of sensors in a WSN is large. As a result, only few nodes (a.k.a. beacons, anchors or pilots) have the known locations with GPS modules. The locations of the remaining sensors (called normal

nodes) are automatically estimated based on the measurements to their neighbors using a localization algorithm.

Generally, localization categorizations are usually made depending on the kind of input data: range-based or range-free; by what methodology the algorithm is performed: centralized or decentralized; and whether the result is gained directly or through iterations: multilateration or successive refinement. Short introductions on these concepts are discussed below.

1.1. Range-based and range-free data

The range-based scheme uses input pair-wise distance data. These measurements can be inferred from such data as received signal strength (RSS), time of arrival (TOA), time of difference of arrival (TDOA), and angle of arrival (AOA) [13]. Measurement error in RSS suffers from fading and shadowing, leading to poor accuracy of localizations using this scheme. Meanwhile, TOA and TDOA also use radio frequency (RF) and acoustic signal as RSS does, but the ranges are estimated based on the time delay of propagation through the environment (TOA) or the time discrepancy of an incoming signal at two different nodes (TDOA) [19]. These measurement methods are more accurate and easier to analyze with the popular Gaussian noise model and, as a trade-off, are more expensive than RSS systems.

For range-free schemes, the input data of localization algorithms is the connectivity information between nodes and their neighbors or the information of who in the communication range [18,17]. A node can also estimate the distances to other nodes by counting the shortest paths (minimum numbers of connections) from itself to other nodes and multiplying this value

* Corresponding author.

E-mail addresses: dangviethung@oslub.khu.ac.kr (V.-H. Dang),
levietduc@oslub.khu.ac.kr (V.-D. Le), yklee@oslub.khu.ac.kr (Y.-K. Lee),
sylee@oslub.khu.ac.kr (S. Lee).

with the mean hop distance [11]. The mean distance may be calculated by averaging all of the hops that make up the shortest paths between anchors. Ranges based on connectivity do not require complicated hardware or high interface power in normal nodes. However, disadvantages of this scheme are: errors of measurements are inevitable; the number of anchors should be high; and the distribution of nodes should be uniform [11].

Since range-based algorithms and most range-free algorithms work on the estimated ranges of pair-wise distances, we do not distinguish between these two schemes. Instead, we concentrate on solving localization problem with the calibrated measurements. Push-pull Estimation (PPE) actually can be seen as a method using the range-based scheme. It is built to deal with not only Gaussian error but with any unbiased error as well. PPE can also solve the problem of biased error measurement input if a de-biasing function is found.

1.2. Centralized and decentralized methodologies

A centralized localization is executed by a single computer. It therefore can easily deal with data combinations and perform complicated computations like matrix operations or eigenvalue, eigenvector calculations of a large matrix [9,15,24]. However, the cost of communications becomes dominant when all of the input data enters into one node, and all of the output data has to exit from only that node. Moreover, the protocols for this kind of method are hard to be implemented. The running time could be long if the complexity of the algorithm is high, especially when the number of nodes in the network becomes large. In other words, the scalability of centralized algorithms is limited. Another variation of the centralized algorithm is related to clustering methods where the whole network is divided into groups, and each group has a central computer to run the algorithm [26]. Yet the disadvantages of this method are difficulties in defining how the groups are divided. The central computer of a group would have strong computing hardware compared to that of a normal node, so it must be carefully deployed so that network communication and computation are effective.

To overcome above disadvantages, researchers have been seeking other methods which can be run distributedly on the simple computers of normal nodes [18,17,11,2,6]. Algorithms of this manner are called decentralized, distributed, pervasive or collaborative localizations. Real-world applications prefer this type of algorithm in which the big load of computations is shared, the calculating time is reduced considerably, and communication bottle-neck is no longer a problem. Our proposed PPE is designed to avoid all of the disadvantages of centralized or clustering-based centralized algorithms. It is purely a distributed method performed by all of the normal nodes in the network.

1.3. Multilateration and successive refinement manner

Multilateration is usually used in distributed algorithms. It takes at least three input range measurements (in 3-D case, four is the least) to calculate a node's coordinates [2]. The ratio of the anchor number to the total node number for this manner is often high because multilateration requires the exact positions of the reference nodes.

Meanwhile, successive refinement obtains the estimated coordinates of all normal nodes through iterations. If it is designed in a decentralized method, each node has to calculate the estimated position, send this information to the network, and receive new estimated positions from other nodes. These tasks are performed repeatedly until convergence. Successive refinement in a distributed methodology is usually more difficult to design, and its convergence is surely a problem [13]. However, it reduces the communication cost and makes the number of anchors not necessary to be as high as in multilateration. In our proposed algorithm, successive updating is used in all sub-stages of the method.

2. Related work

Only few range-free methods like RF-based fingerprint matching [25] and grid-scanning [18,3] do not use range estimation. The other range-free methods, to improve accuracy, estimate distances by counting the hops between nodes and then using these estimates as their inputs. For these ones, distance-vector-based (DV-based) positioning [11] is one good way for approximating the ranges. Apparently, pair-wise distance is quite necessary not only with range-based schemes but also in range-free schemes. In both of these schemes, the statistical model of pair-wise measurement is usually assigned to the very popular Gaussian model. This model represents the most unpredicted noise (maximum entropy) and has a parametric function which is easy for analysis. However, it is only appropriate for TOA, TDOA, and AOA measurements. In WSNs, where RF signals are always available, RSS is likely the cheapest and most convenient means for range-based schemes. Its error model, however, is not typical Gaussian but a log-normal distribution, or Gaussian in the log-domain [13,2,6,7]. In [7], although Malaney does not give solutions, he presents how log-normal fading models influence location accuracy and also gives the mathematical analysis for the combination model of multipath fading and log-normal power shadowing. Another technique using parametric channel models is proposed in [21] to reduce error by using the information of many parameters to refine the node locations. Recently, an approach in [10] has used least square (LS) and maximum likelihood (MLE) algorithms to locate an unknown-coordinate node with references to anchors.

The mentioned localizations are either “centralized and successive refinement” or “decentralized and non successive refinement”. Since successive refinement methods usually give higher accuracy and decentralized methods are more practical, the combination has gained more and more research interest. Patwari and Costa first propose a learning-based algorithm [14], a candidate with the ability to find the eigenvalues and eigenvectors of a sparse and symmetric matrix. Then, they introduce a distributed weighted-multidimensional scaling approach (dwMDS [2]) developed from classical multidimensional scaling (MDS), a centralized and more complex algorithm. In fact, MDS is such a novel tool that many of its variations have been developed for node localization in WSNs. Iterative MDS (IT-MDS) and simulated annealing MDS (SA-MDS) discussed in [1] are two of these variations in which simulated annealing is a famous method imitating the metal cooling process to find the optimal state. Fastmap and MDS are combined in [4] to give a two-stage algorithm in which Fastmap provides the initial coarse input and MDS does the distributed gradient descent for more accuracy like in [2]. Generally, MDS variations require a big load of computation and result in poor accuracy, so the scalability of the methods is still an issue. A distributed MLE with better accuracy [15] has been introduced, but this increased accuracy occurs only when the algorithm has good first guess input data (raw estimation). In [6], another method uses MDS to calculate the initial data and then uses MLE to produce the refined result of localization, making the computation and communication costs even higher. In this paper, we introduce a distributed successive localization. At each iteration on a normal node, only few simple computations are needed. Moreover, the communication cost can be reduced remarkably with an insignificant accuracy trade-off. One remarkable point is that PPE is unlike other methods which base on available algorithms, it needs to be analyzed fully. The analyses in later sections prove that PPE can obtain the global convergence even it is performed distributedly. It is a convergent and robust algorithm with any unbiased measurement input. Therefore, it can deal with not only Gaussian model but the log-normal model with a de-biasing function as well.

The rest of the paper is organized as follows. The main analyses of PPE are in Section 3 where the annotations are

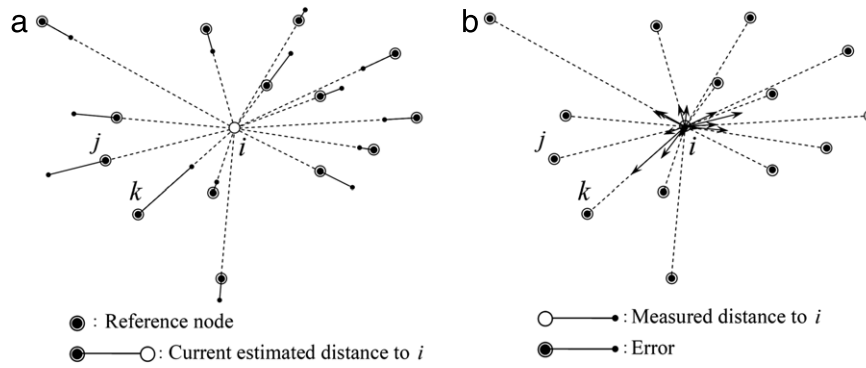


Fig. 1. (a) Node i and measurement errors from i to its reference nodes. (b) The errors are modeled as pull–push forces.

introduced and the analyses of convergence, distributed ability, scalability, and communication cost reducing ability are presented. Section 4 includes the simulations for the purposes of verifying the analyses and comparing the results with those of other contemporary successive refinement approaches. Finally, we present our conclusions in Section 5. Besides, a list of main parameters for descriptions and analyses is available in the Appendix at the end of this paper for the convenience of reading.

3. Push–pull Estimation (PPE)

3.1. Annotation and description of PPE

Our proposed algorithm [5] is based on geometry in which the errors of measurements are modeled into pushing and pulling forces. The influence of these forces leads a node to the point where all of the forces are balanced or, in other words, where the errors are minimized and canceled out through the averaging mechanism. The algorithm is named the Push–pull Estimation because of this original concept. PPE is designed to meet the basic requirement that if the range measurement is shorter than the calculated distance between two nodes, these two nodes pull each other, and vice versa. In addition, the force magnitude should be a monotone function of the discrepancy between the measured and the calculated distances in order for PPE to work and converge. The bigger is the discrepancy, the higher is the influence it gives to reduce itself. This idea is demonstrated in Fig. 1 where node i needs to update its coordinates to the balanced position. Node j causes a pushing force on i because the range measurement between i and j is longer than the current calculated distance between them, and contrarily, node k exerts a pulling force on i . It should be noted that in the concept of this model, we give normal nodes the moving ability and consider location updating as a process in which normal nodes move under the influence of forces caused by their neighbors. When a normal node attempts to locate its balanced position, it considers the other nodes to be still.

Assuming that the total node number of a 2-D deployed sensor network is N , $N = n + m$ where n is the number of normal nodes and m is the number of anchors, $m \ll n$. Let $\mathbf{x} = \{x_i : i = 1, \dots, N\}$, $x_i \in \mathbf{R}^2$, be the true coordinate vector of normal sensors $\{x_i : i = 1, \dots, n\}$ and anchors $\{x_i : i = (n + 1), \dots, N\}$. The true distance d_{ij} is the Euclidean distance between i and j ,

$$d_{ij} = \|x_j - x_i\|; \quad i, j = 1, \dots, N, \quad (1)$$

so, $d_{ii} = 0$ and $d_{ij} = d_{ji}$.

We used the range-based scheme for PPE input by assuming that with a certain methodology, unbiased measurements from a node to its neighbors and several anchors are available. This is normal in most range-based localized WSNs. The ranges can also be estimated through hops [11,22], and the anchors may have a

large broadcasting range. Then the pair-wise measurement or the measured distance between node i and j is $\delta_{ij} : i, j = 1, \dots, N$; $\delta_{ij} = \delta_{ji}$, $\delta_{ii} = 0$ and $\delta_{ij} = 0$ if the range from i to j is too large, or out of range. Since the error of measurement is always proportional to the real distance, δ_{ij} has the following form [4]

$$\delta_{ij} = d_{ij} + d_{ij} \cdot \text{noise}_{ij}, \quad i, j = i, \dots, N \quad (2)$$

where $\text{noise}_{ij} \sim \mathcal{K}(0, \sigma_n^2)$, some zero-mean distribution. We also define the current estimated coordinate vector to be $\tilde{\mathbf{x}} = \{\tilde{x}_i : i = 1, \dots, N\}$, $\tilde{x}_i \in \mathbf{R}^2$, certainly we always have $\tilde{x}_j = x_j$, $j = (n + 1), \dots, N$. The localization problem can be succinctly stated: the input data of the algorithm are measurements δ_{ij} and anchors' locations $\{x_j : j = (n + 1), \dots, N\}$; the algorithm will give an initial $\tilde{\mathbf{x}}$ and will update each \tilde{x}_i of $\tilde{\mathbf{x}}$ so that the difference between $\tilde{\mathbf{x}}$ and \mathbf{x} is as small as possible, $\tilde{\mathbf{x}}$ then is the solution to the problem. Let the current calculated pair-wise ranges be \tilde{d}_{ij} , then

$$\tilde{d}_{ij} = \|\tilde{x}_j - \tilde{x}_i\|, \quad i, j = 1, \dots, N. \quad (3)$$

Now the distributed PPE algorithm for a normal node i is presented with three phases. In each phase node i uses iterations to produce a balanced position by updating position \tilde{x}_i with the two following equations:

$$\vec{F}_i^{(p)} = \frac{1}{M_i^{(p)}} \sum \vec{f}_{ij}^{(p)}, \quad (4)$$

$$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(p)} \vec{F}_i^{(p)}. \quad (5)$$

$\alpha^{(p)}$ in (5) is the movement rate and $\vec{F}_i^{(p)}$ is the mean-force caused by a set of $M_i^{(p)}$ related nodes. These parameters vary in different phase p of PPE.

– Phase 1: Raw estimation:

Node i needs at least three different beacon positions and the measurements from itself to these beacons. By computing (4) and (5), i gets its raw estimated location where the sum of forces caused by the related beacons is balanced. At each iteration, these forces are

$$\vec{f}_{ij}^{(1)} = (\tilde{d}_{ij} - \delta_{ij}) \vec{e}_{ij}, \quad (6)$$

where j is the index of beacons related to i , and \vec{e}_{ij} is the unit vector pointing from \tilde{x}_i to \tilde{x}_j to indicate the direction of \vec{f}_{ij} . The initial position of \tilde{x}_i is chosen as the mean position of these beacons. Choosing the initial position is empirical and will be mentioned later in Section 4.1. $M_i^{(1)}$ is the number of beacons related to i .

– Phase 2: Pre-refinement:

After normal nodes complete the raw estimation phase, node i takes the measurements to all of its related nodes, consisting of related neighbors and beacons, and their current updated positions

for determining its new balanced position. For normal neighbor j , \tilde{x}_j is the current estimated position and for related beacon j , it is the actual position. At each round the individual forces are

$$\vec{f}_{ij}^{(2)} = (\tilde{d}_{ij} - \delta_{ij}) \vec{e}_{ij}^{(2)}. \quad (7)$$

For both phase 2 and phase 3, after a certain number of iterations, node i must send out its current updated position to its normal neighbors and get the new updated positions of the neighbors for better location computing.

– Phase 3: Refinement:

This phase is used last to improve result accuracy because the error of measured distance is proportional to the actual distance. It has the same behavior as phase 2 with the same related nodes, or $M_i^{(3)} = M_i^{(2)}$. The difference is the way node i calculates the forces caused by all of its related nodes. $\vec{f}_{ij}^{(3)}$ is obtained by dividing $\vec{f}_{ij}^{(2)}$ by \tilde{d}_{ij} ,

$$\vec{f}_{ij}^{(3)} = \left(\frac{\tilde{d}_{ij} - \delta_{ij}}{\tilde{d}_{ij}} \right) \vec{e}_{ij} = \left(1 - \frac{\delta_{ij}}{\tilde{d}_{ij}} \right) \vec{e}_{ij}. \quad (8)$$

j in phase 3 is the same as that in phase 2, the index of all nodes related to i . The stop condition for all three phases is when the magnitude of the mean-force on i is less than some small positive value or when a given maximum number of iterations is reached. Note that $\tilde{d}_{ij} \neq 0$ in all three phases, $\tilde{d}_{ij} = 0$ means that i does not have the measured distance to j and so will not use j as a related node for updating its position.

For convenience of description and further analysis, we add the following common formula

$$\vec{f}_{ij}^{(p)} = \left(\frac{\tilde{d}_{ij} - \delta_{ij}}{L_{ij}} \right) \vec{e}_{ij}, \quad (9)$$

where

$$L_{ij} = \begin{cases} 1 & \text{for phase 1 and phase 2} \\ \tilde{d}_{ij} & \text{for phase 3.} \end{cases} \quad (10)$$

Clearly, if $\delta_{ij} < \tilde{d}_{ij}$, then \vec{f}_{ij} , having the same direction with vector \vec{e}_{ij} , will be a pulling force given by node j on node i ; and vice-versa, if $\delta_{ij} > \tilde{d}_{ij}$, \vec{f}_{ij} will be a pushing force. If $\delta_{ij} = \tilde{d}_{ij}$, the force of j on i is zero.

This force model is the core idea of the algorithm. When a normal node, say node i , is at a position where the forces around it are not balanced, i will move under the force influence to a new position where the sum-force tends to be weaker. While i is moving, at any new position, the sum-force changes in both of magnitude and direction, so the path of movement is actually a curve. Since we cannot make a continuous curve, we will use step-wise movements over iterations to update the estimation. In addition, $\alpha^{(p)} \vec{F}_i^{(p)}$ must be a quantity in the same unit as that of d_{ij} or δ_{ij} . We can define $\alpha^{(p)}$ with a more specific formula

$$\begin{cases} \alpha^{(1)} = \alpha_1 \\ \alpha^{(2)} = \alpha_2 \bar{d}_{ij} \\ \alpha^{(3)} = \alpha_3 \bar{d}_{ij} \end{cases} \quad (11)$$

where α_i is a non-unit quantity and is small enough to guarantee convergence; and \bar{d}_{ij} is the mean value of the distances measured from i to its neighbors and can be replaced by other parameters provided that the replaced ones have the same unit of length. Determining the value of $\alpha^{(p)}$ so that the step is small enough to guarantee convergence is difficult. If $\alpha^{(p)}$ is too small, too

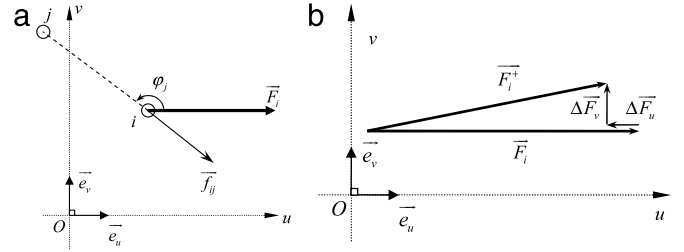


Fig. 2. (a) The mean-force in coordinate system uOv . (b) The mean-force decreases in magnitude when a node moves along it.

many iterations are needed. If it is too large, the algorithm may not converge. There are a few approaches for adjusting $\alpha^{(p)}$ to overcome this dilemma. For consistency with the goal of low cost, we choose the *line search* technique by doubling $\alpha^{(p)}$ until $\|\vec{F}_i^{(p)}\|$ starts to increase.

For evaluating the accuracy of PPE in later sections, we use the root mean square of the position errors.

$$\text{RMS} = \sqrt{\frac{1}{n} \sum_k \|\tilde{x}_k - x_k\|^2}, \quad (12)$$

where k is the index for a normal node.

3.2. PPE's convergence analysis

We analyze PPE's convergence by proving the following statements:

(i) "In all three phases, a node will always find a balanced position via the force mechanism".

This statement implies that the force described in (9) converges the location of a normal node i to a balanced point, and therefore i would never get stuck in infinite loop. We decompose the mean-force \vec{F}_i ($\vec{F}_i = F_i^{(1,2)}$) into two orthogonal sub-components \vec{F}_u and \vec{F}_v in a coordinate system uOv in which \vec{e}_u and \vec{e}_v are the basis vectors. In order to simplify the analysis, we choose $(\vec{F}_i, \vec{Ou}) = 0$

and let $\varphi_j = (\vec{F}_i, \vec{ij})$ (see Fig. 2). For phase 1 and phase 2, the mean-force is rewritten:

$$\vec{F}_i = \frac{1}{M_i^{(p)}} \sum_j (\tilde{d}_{ij} - \delta_{ij}) \vec{e}_{ij}, \quad (13)$$

$$\vec{F}_i = \vec{F}_u + \vec{F}_v, \quad (14)$$

$$\begin{aligned} \vec{F}_i &= \frac{\vec{e}_u}{M_i^{(p)}} \sum_j \left(\left(\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} - \delta_{ij} \right) \cos \varphi_j \right) \\ &\quad + \frac{\vec{e}_v}{M_i^{(p)}} \sum_j \left(\left(\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} - \delta_{ij} \right) \sin \varphi_j \right). \end{aligned} \quad (15)$$

Now we analyze how the mean-force changes when i moves along \vec{F}_i 's direction a small distance that φ_j is considered to be unchanged. It should be noted that although the v -component $\vec{F}_v = 0$, when i moves along \vec{F}_i , \vec{F}_i becomes \vec{F}_i^+ , and the v -

component of \vec{F}_i^\dagger is not zero. Partial derivation of Eq. (15) reveals

$$\frac{\partial \vec{F}_i}{\partial u_i} = \frac{\vec{e}_u}{M_i^{(p)}} \sum_j \left(\frac{(u_i - u_j) \cos \varphi_j}{\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2}} \right) + \frac{\vec{e}_v}{M_i^{(p)}} \sum_j \left(\frac{(u_i - u_j) \sin \varphi_j}{\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2}} \right), \quad (16)$$

$$\frac{\partial \vec{F}_i}{\partial u_i} = \left(\frac{1}{M_i^{(p)}} \sum_j -\cos^2 \varphi_j \right) \vec{e}_u + \left(\frac{1}{M_i^{(p)}} \sum_j -\cos \varphi_j \sin \varphi_j \right) \vec{e}_v. \quad (17)$$

Eq. (17) means that when i moves along \vec{F}_i a very small distance Δu ($\Delta u > 0$), \vec{F}_i changes by an amount of $\Delta \vec{F}_i$, which is composed by two orthogonal components $\Delta \vec{F}_u$ and $\Delta \vec{F}_v$ (see Fig. 2),

$$\Delta \vec{F}_i = \Delta \vec{F}_u + \Delta \vec{F}_v. \quad (18)$$

$$\Delta \vec{F}_i = \left(\frac{1}{M_i^{(p)}} \sum_j -\cos^2 \varphi_j \right) \Delta u_i \vec{e}_u + \left(\frac{1}{M_i^{(p)}} \sum_j -\cos \varphi_j \sin \varphi_j \right) \Delta u_i \vec{e}_v. \quad (19)$$

The new updated mean-force on i becomes

$$\vec{F}_i^\dagger = \vec{F}_i + \Delta \vec{F}_i. \quad (20)$$

Because of the fact that $\sum_j -\cos^2 \varphi_j < 0$, the inner product $\langle \Delta \vec{F}_u, \vec{e}_u \rangle < 0$, or $\Delta \vec{F}_u$ always has a counter-direction to \vec{F}_i . In addition, $\|\Delta \vec{F}_v\| \ll \|\vec{F}_i\|$. As a result, for phase 1 and phase 2 (see Fig. 2(b)):

$$\|\vec{F}_i^\dagger\| \approx \|\vec{F}_i + \Delta \vec{F}_u\| < \|\vec{F}_i\|. \quad (21)$$

In the same manner, in phase 3, we infer the change amount of $\vec{F}_i^{(3)}$,

$$\frac{\partial \vec{F}_i^{(3)}}{\partial u_i} = \frac{1}{2M_i^{(3)}} \left(\sum_j \left(-\frac{\delta_{ij}}{\tilde{d}_{ij}^2} \right) \cos^2 \varphi_j \right) \vec{e}_u + \frac{1}{2M_i^{(3)}} \left(\sum_j \left(-\frac{\delta_{ij}}{\tilde{d}_{ij}^2} \right) \sin \varphi_j \cos \varphi_j \right) \vec{e}_v. \quad (22)$$

The u -component $\Delta \vec{F}_u^{(3)}$ is also opposite in direction with $\vec{F}_i^{(3)}$. Therefore, (21) holds for all three phases of PPE. This means that when i moves under PPE's force effect, the mean-force's magnitude gets smaller and smaller. As a result, i 's location eventually converges to a balanced position.

One can easily see that PPE is basically an optimization method for a normal node in which the objective function to minimize is the length of the mean-force,

$$\tilde{x}_i = \arg \min_{\tilde{x}_i} \left\| \vec{F}_i^{(p)} \right\|. \quad (23)$$

Aiming at a low-cost algorithm, we have avoided much of the complicated load of the gradient descend method which provides

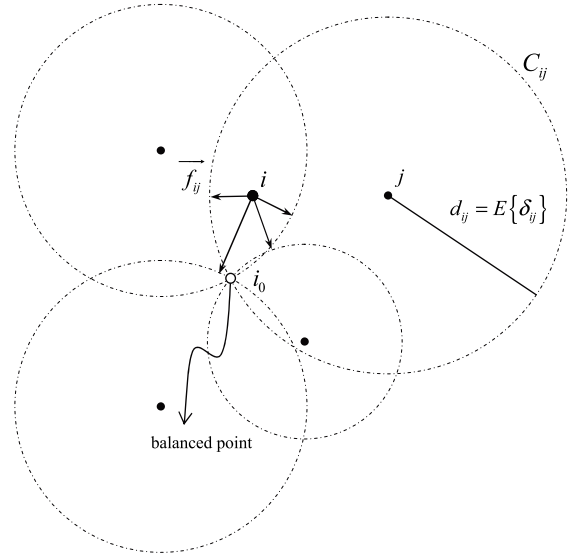


Fig. 3. When the measurements have no error and the reference nodes' locations are known, the balanced point of a node is its actual position.

the optimal direction of movement. By letting a normal node gradually move along the mean-force, our method is simpler and involves only few simple computations. The proof above guarantees the minimization.

(ii) "Estimation of a normal node in phase 1 is unbiased".

In this phase, the reference nodes of a normal node i are the anchors. Since $E\{\delta_{ij}\} = d_{ij}$, the expectation of the mean-force is

$$E\left\{ \vec{F}_i^{(1)} \right\} = \frac{1}{M_i^{(1)}} \sum_j (\tilde{d}_{ij} - d_{ij}) \vec{e}_{ij} \quad (24)$$

where j is the index of the known-location nodes. This equation implies that the expectation of the mean-force is built upon the correct measurements from the known-location nodes. Each individual force f_{ij} tries to move i to the circle C_{ij} , having the radius d_{ij} , to minimize its own magnitude (see Fig. 3). As the result, i 's position converges to the intersection point i_0 of these circles C_{ij} , or

$$E\{\tilde{x}_i\} = x_i. \quad (25)$$

(iii) "Estimations of a normal node in phase 2 and phase 3 are unbiased".

In these two phases, every normal node has its raw location estimation and uses all of its reference nodes to better the estimation. Since the algorithm mainly deals with probability issues, we not only consider the force effects to be specifically caused by the nodes but also consider the force effects caused by sub-areas. We assume that the whole deployed area is divided into small sub-areas, in each of which the node number is large and the node distribution is uniform. Fig. 4 illustrates this idea: the actual positions are the black round nodes, e.g., k_0 , while the current estimated positions are the smaller black square nodes, e.g., k . The solid links between them are the current errors of estimations. What we want to do here is to prove that even with these errors of estimations, the balanced position of node i under PPE mechanism is its actual position i_0 .

Let the mean force on i caused by the area S_K be

$$\vec{F}_{iK} = \frac{1}{n_K} \sum_k \left(\frac{\tilde{d}_{ik} - \delta_{ik}}{L_{ik}} \right) \frac{\vec{iK}}{\tilde{d}_{ik}} \quad (26)$$

$$E\left\{ \vec{F}_{iK} \right\} = E\left\{ \left(\frac{\tilde{d}_{ik} - d_{ik} - d_{ik} \cdot \text{noise}_{ik}}{L_{ik}} \right) \frac{\vec{iK} + \vec{Kk}}{\tilde{d}_{ik}} \right\}. \quad (27)$$

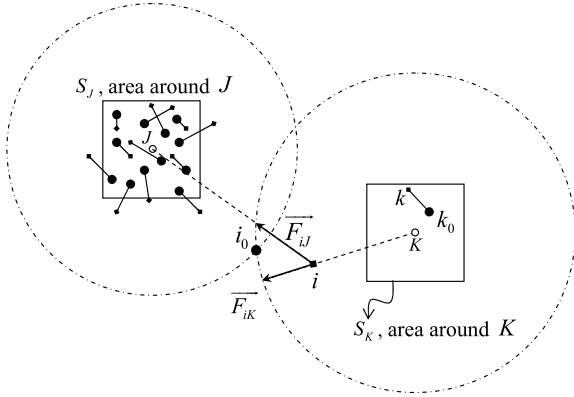


Fig. 4. The force effect of a sub-area on a node is equivalently the force effect caused by the center position of this sub-area with the correct measurement.

K is the center point of S_K , and n_k in (26) is the number of nodes in S_K . In (27), $noise_{ik}$ and \vec{Kk} are independent of each other and of d_{ij} , \tilde{d}_{ij} . $E\{\vec{Kk}\} = 0$ since the estimation of the previous phase is unbiased. Eq. (27) is rewritten as

$$E\{\vec{F}_{ik}\} = E\left\{\left(\frac{\tilde{d}_{ik} - d_{ik}}{\tilde{d}_{ik}L_{ik}}\right)\right\} \vec{iK} \quad (28)$$

$$E\{\vec{F}_{ik}\} = E\left\{\left(\frac{ik - i_0k_0}{\tilde{d}_{ik}L_{ik}}\right)\right\} \vec{iK}. \quad (29)$$

Assume that the estimation errors of nodes in S_k are small so that k still has a uniform distribution in S_K , or $E\{ik\} = E\{i_0k_0\}$. If $ik = i_0k_0$, then $E\{ik_0 - i_0k_0\} = 0$. Since $\tilde{d}_{ik}L_{ik}$ is a positive and bounded value, $E\{\vec{F}_{ik}\} = 0$. If $ik > i_0k_0$, then $E\{ik_0 - i_0k_0\} > 0$, and $E\{\vec{F}_{ik}\}$ becomes a pull force with the direction of \vec{iK} . Conversely, if $ik < i_0k_0$, then $E\{\vec{F}_{ik}\}$ becomes a push force. Hence, the balanced point of i belongs to the circle $C(K, i_0K)$. Combining the effects of other sub-areas on i , we find that i_0 becomes the unique balanced position of i . With the previous proof that PPE leads i to a balanced point, we can conclude that the estimations of i in phase 2 and phase 3 are unbiased.

In these two phases, a normal node acts as if it is affected by forces caused by groups of reference nodes, where each group plays the role of a beacon. With more reference beacons, phase 2 has a better accuracy of estimation than does phase 1 (see Section 3.3). In addition, it should be noted that the assumption that k has a uniform distribution in S_K is not a tight one. However, through iterations, each node updates and achieves a better and better estimation, so the assumption becomes tighter and tighter. Consequently, the expectation of the balanced point of a normal node i is the very i_0 , and PPE drags i toward i_0 even when the current estimations of the reference nodes are experiencing errors. In other words, one node's updating is independent of the others' and global convergence is obtained. This means that the distributed PPE is a robust convergent algorithm.

3.3. Necessity of PPE's phases

Raw estimation, i.e. initial input data, is the same requirement for most successive localizations, unless the Procrustes algorithm is used [6,4]. Procrustes is the method used in the last module of localization to fix a flipped or rotated result. However, it usually needs all of the node location estimations, leading to a centralized mechanism and the associated problems, which all

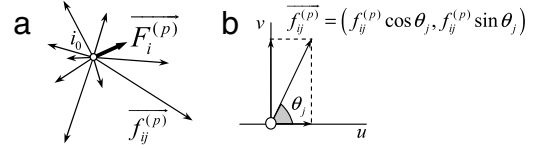


Fig. 5. (a) The mean-force is not a zero vector at the correct position because measurements have errors and the number of reference nodes is limited. (b) A force is decomposed into horizontal and vertical components.

distributed approaches aim to avoid. The raw estimation can be any one of the low-cost algorithms so far. However, multilateral localizations have either complicated calculation (localizing the most likely area) or a high number of anchors [18,21,14,23,8,16]. Contemporary successive algorithms are surely more complex; in [6], MDS is used to obtain this raw estimation. Clearly, PPE's phase 1 plays an important role for the later phases based only on simple computations. Moreover, by using phase 1, we can utilize the same hardware and software modules because the structures of all of three phases are similar. This makes the method more practical.

In all three phases, if the reference node number is infinitely large, the node's balanced position will be at its actual position. Unfortunately, the limited reference node number $M_i^{(p)}$ and the errors of measurements make the mean-force non-zero at the correct position i_0 , drive i out of i_0 , and rise PPE's error (see Fig. 5(a)). Obviously, the larger is the variance of the mean-force's magnitude, the larger is the error in the final result. Therefore, to compare the performances of the three phases, we examine the variance of $\|\vec{F}_i^{(p)}\|$ at position i_0 when all of the reference nodes are at the correct positions.

Let $\|\vec{F}_i^{(p)}\| = F_i^{(p)}$, $\|f_{ij}^{(p)}\| = f_{ij}^{(p)}$ and decompose each individual force $\vec{f}_{ij}^{(p)}$ into components as in Fig. 5(b), so $\vec{f}_{ij}^{(p)} = (f_{ij}^{(p)} \cos \theta_j, f_{ij}^{(p)} \sin \theta_j)$. Remarking that $E\{f_{ij}^{(p)}\} = 0$, $E\{F_i^{(p)}\} = 0$, (unbiased measurement error and unbiased estimation respectively), and $f_{ij}^{(p)}$ and θ_j are two independent variables, we obtain

$$\vec{F}_i^{(p)} = \left(\frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \cos \theta_j, \frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \sin \theta_j \right) \quad (30)$$

$$\text{var}(F_i^{(p)}) = E\left\{(F_i^{(p)})^2\right\} - E^2\{F_i^{(p)}\} \quad (31)$$

$$\text{var}(F_i^{(p)}) = E\left\{\left(\frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \cos \theta_j\right)^2 + \left(\frac{1}{M_i^{(p)}} \sum_j f_{ij}^{(p)} \sin \theta_j\right)^2\right\} \quad (32)$$

$$\text{var}(F_i^{(p)}) = \frac{\text{var}(f_{ij}^{(p)})}{M_i^{(p)}}. \quad (33)$$

Eq. (33) implies that the variance of $F_i^{(p)}$ does not depend on the directions of the forces caused by neighbors, but only on the reference node number and the variance of the individual force. It also guarantees that phase 2 is necessary to improve phase 1's result, which is obtained only with anchors ($M_i^{(1)} < M_i^{(2)}$).

In order to show the necessity of phase 3 over phase 2, we compare the variances of the individual forces in these two phases

because the number of reference nodes in these phases are the same, $M_i^{(2)} = M_i^{(3)}$. From (4) and (11), we have

$$\begin{cases} \text{var}(f_j^{(2)}) = \text{var}(\alpha_2(\delta_{ij} - d_{ij})) = \text{var}(\alpha_2 d_{ij} \cdot \text{noise}_{ij}) \\ \text{var}(f_j^{(3)}) = \text{var}\left(\alpha_3 \bar{d}_{ij} \cdot \frac{\delta_{ij} - d_{ij}}{d_{ij}}\right) = \text{var}(\alpha_3 \bar{d}_{ij} \cdot \text{noise}_{ij}). \end{cases} \quad (34)$$

Obviously, in each phase, the balanced position of a node is the same with any α_i provided that α_i is small enough so that PPE can obtain convergence. Therefore the comparison is meaningful only when the forces have equivalent magnitudes, or

$$\sum_j \alpha_2 d_{ij} \cdot \text{noise}_{ij} \approx \sum_j \alpha_3 \bar{d}_{ij} \cdot \text{noise}_{ij} \Leftrightarrow \alpha_2 \approx \alpha_3. \quad (35)$$

Then,

$$\begin{aligned} \text{var}(f_j^{(2)}) &= \alpha_2^2 \left[E \left\{ (d_{ij} \cdot \text{noise}_{ij})^2 \right\} - E^2 \left\{ d_{ij} \cdot \text{noise}_{ij} \right\} \right] \\ &= \alpha_2^2 \left[E \left\{ d_{ij}^2 \right\} E \left\{ (\text{noise}_{ij})^2 \right\} - 0 \right] \end{aligned} \quad (36)$$

$$\begin{aligned} \Leftrightarrow \text{var}(f_j^{(2)}) &= \alpha_2^2 (\text{var}(d_{ij}) + E^2 \{d_{ij}\}) \text{var}(\text{noise}_{ij}) \\ &= \alpha_2^2 E^2 \{d_{ij}\} \text{var}(\text{noise}_{ij}) \\ &\quad + \alpha_2^2 \text{var}(d_{ij}) \cdot \text{var}(\text{noise}_{ij}). \end{aligned} \quad (37)$$

Meanwhile,

$$\text{var}(f_j^{(3)}) = \alpha_2^2 \bar{d}_{ij}^2 \text{var}(\text{noise}_{ij}) = \alpha_2^2 E^2 \{d_{ij}\} \text{var}(\text{noise}_{ij}). \quad (38)$$

Eqs. (37) and (38) lead to

$$\text{var}(f_j^{(3)}) < \text{var}(f_j^{(2)}). \quad (39)$$

The variance of the mean-force is reduced in the refinement phase. Consequently, the result of estimation in phase 3 is better than that in phase 2 when the measurement error is proportional to the real distance. Confirmations are conducted via simulation in Section 4.3.

3.4. Distributed ability and scalability

In any of the three mentioned phases, PPE works distributedly in the way that the computation load of localization is divided into sub-tasks performed by the normal nodes of the network. Statements proved in Section 3.2 assure the convergence of the whole result, even if the reference positions are not correct. Hence, normal nodes do not require synchronized updating while the localization is being performed. In other words, PPE is a distributed algorithm.

One important point is that the lack of minor information can be accepted by PPE. Removed or added nodes will not affect much a node's position updating as long as the number of these nodes is small compared to the number of reference neighbors. The reason is that each individual force contributes to the mean-force an amount proportional to $1/M_i^{(p)}$. Some nodes, including newly added ones, even do not need to obtain the distances to the beacons beforehand because they can use the estimated positions of the reference neighbors. Besides, each node only uses the information of its nearest anchors and neighbors, so the scalability of a network using PPE is assured.

In any distributed successive refinement localizations, a node needs to know its neighbors' current positions in order to obtain a higher estimation at every iteration. The communication cost would significantly increase, whereas network designers continually try to reduce and replace it with computation cost. Fortunately, PPE is a good way to soothe the burden of communication cost. Proven statements guarantee that a node's updated result is not affected much by those of others. Hence, it

is unnecessary for a node to send out requests to its neighbors for reference updated positions after every iteration. Instead, the communication cost can be reduced when a normal node sends the request task once after every R internal iterations on it, or in other words, the task is skipped $R - 1$ times. We call R the *cycle request number*. Consequently, the communication cost decreases by R -fold.

$$\text{PPE_Communication_Cost} = \frac{\text{C.Cost}}{R}, \quad (40)$$

where C.Cost is the total communication cost of the network when $R = 1$. This again verifies PPE's distributed ability and scalability for practical implementation with light computation and communication. A simulation set in Section 4.1 shows how little a node's iteration result depends on those of the others; and another in Section 4.3 illustrates the slight trade-off of accuracy for significant communication cost.

3.5. De-biasing function and RSS scheme

The noise model in this article does not have to be a specific one which can be represented by a mathematical formula. The only constraint of the noise model is a zero-mean distribution or unbiased measurements.

Suppose we have a function $g(\cdot)$ that changes the distribution of variable δ_{ij} into an unbiased distribution of variable $g(\delta_{ij})$ and $g(\cdot)$ is a monotone increasing function so that PPE can correctly define the direction of the force, the formula (6), (7) and (8) now can be replaced by

$$\vec{f}_{ij}^{(1)} = \vec{f}_{ij}^{(2)} = \left(g(\delta_{ij}) - g(\tilde{d}_{ij}) \right) \vec{e}_{ij} \quad (41)$$

and

$$\vec{f}_{ij}^{(3)} = \left(\frac{g(\delta_{ij}) - g(\tilde{d}_{ij})}{g(\tilde{d}_{ij})} \right) \vec{e}_{ij}. \quad (42)$$

Phase 3 with (42) is necessary only when the error of $g(\delta_{ij})$ is proportional to $g(d_{ij})$. In reality, RSS power decays proportionally to a negative exponent of the distance. As a result, the power error measured at a receiver is modeled to the log-normal distribution, $P_{ij} \sim N(E\{P_{ij}\}, \sigma_p^2)$. We will find the de-biasing function $g(\cdot)$ for this case:

$$P_{ij} = P_0 - 10 \log_{10} \left(\frac{\delta_{ij}}{d_0} \right), \quad (43)$$

$$P_{ij} = E\{P_{ij}\} + Pnoise. \quad (44)$$

P_{ij} is the received power in decibels at node i when node j transmits to i and $Pnoise \sim N(0, \sigma_p^2)$ while P_0 is the received power, also in decibels, from a node at a reference distance d_0 . In this article, we do not focus on the physical parameter channel-loss, which is included in the Gaussian variable $Pnoise$ (see Appendix of [4]), and just use $Pnoise$ for the simulation of the log-normal measurement model. Rewrite (44):

$$10 \log_{10}(\delta_{ij}) = 10 \log_{10}(d_{ij}) - Pnoise. \quad (45)$$

Even though δ_{ij} does not have an unbiased distribution over d_{ij} , $10 \log_{10}(\delta_{ij})$ does over $10 \log_{10}(d_{ij})$. Therefore the de-biasing function in PPE to deal with the log-normal distribution is $g(d) = 10 \log_{10}(d)$. Actually, the function $g(d)$ does not need to be exactly $10 \log_{10}(d)$, but any of the log variations providing that $g(d)$ is a monotone increasing function and $g(\delta_{ij}) = g(d_{ij}) + t \cdot N(0, \sigma_p^2)$, $t \neq 0$. Then,

$$g(d) = a \log_{10}(d), \quad a \in R^+. \quad (46)$$

For a log-normal RSS scheme, the measurement model is (from (45))

$$\delta_{ij} = d_{ij} 10^{P_{noise}/10}, \quad (47)$$

and only the first two phases with (41) are used.

3.6. Details of PPE

As a consequence of all PPE's characteristics and the related analysis, PPE for a normal node i now can be described in details. In this description, $\vec{f}_{ij}^{(p)}$ is defined in (41) and (42); $g(\cdot)$ is the de-biasing function. We choose $g(d) = d$ if the distance measurement is unbiased and $g(d) = \log_{10} d$ if it has a log-normal distribution. R is the cycle request number.

Algorithm 1: Phase 1: Raw estimation

+ Send out requests for nearest anchors' coordinates x_j (at least three) and the measurements to each.

+ Set the initial position, $\tilde{x}_i = \frac{1}{M_i^{(1)}} \sum_j x_j$.

+ Set the initial moving step $\alpha^{(1)}$.

+ $k^{(1)} = 0$, $k^{(1)}$ is the iteration index.

repeat

$k^{(1)} = k^{(1)} + 1$;

$\vec{F}_i^{(1)} = \frac{1}{M_i^{(1)}} \sum_j \vec{f}_{ij}^{(1)}$;

 (j is the index of the nearest **beacons**).

if $F_i^{(1)}$ still decreases over the iterations **then**

 Double $\alpha^{(1)}$;

else

 Halve $\alpha^{(1)}$;

 Stop updating $\alpha^{(1)}$;

$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(1)} \vec{F}_i^{(1)}$;

until ($k^{(1)} > k_{\max}^{(1)}$) OR ($\|\vec{F}_i^{(1)}\| < \epsilon$);

Algorithm 2: Phase 2: Pre-refinement

+ $\alpha^{(2)} = \alpha^{(1)}$;

+ $k^{(2)} = 0$, $k^{(2)}$ is the iteration index.

repeat

if $\text{mod}(k^{(2)}, R) == 0$ **then**

 Request the current updated coordinates of the neighbors.

 Send out the current self updated position \tilde{x}_i to the neighbors.

$k^{(2)} = k^{(2)} + 1$;

$\vec{F}_i^{(2)} = \frac{1}{M_i^{(2)}} \sum_j \vec{f}_{ij}^{(2)}$;

 (j is the index of all **reference nodes**).

$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(2)} \vec{F}_i^{(2)}$;

until ($k^{(2)} > k_{\max}^{(2)}$) OR ($\|\vec{F}_i^{(2)}\| < \epsilon$);

4. Experiment results and discussions

We conduct four main experiments. The first experiment focuses on PPE's convergence confirmation; the second tests the accuracy enhancement of the refinement phase compared with that of the pre-refinement phase. The third experiment is about PPE's ability to reduce communication cost, while the last one analyzes the performance comparison of PPE with dwMDS and MLE. We mainly examine a group of nodes to highlight PPE's

Algorithm 3: Phase 3: Refinement

+ $\alpha^{(3)} = \alpha^{(2)}$;

+ $k^{(3)} = 0$, $k^{(3)}$ is the iteration index.

repeat

if $\text{mod}(k^{(3)}, R) == 0$ **then**

 Request the current updated coordinates of the neighbors.

 Send out the current self updated position \tilde{x}_i to the neighbors.

$k^{(3)} = k^{(3)} + 1$;

$\vec{F}_i^{(3)} = \frac{1}{M_i^{(3)}} \sum_j \vec{f}_{ij}^{(3)}$;

 (j is the index of all **reference nodes**).

if $F_i^{(3)}$ still decreases over the iterations **then**

 Double $\alpha^{(3)}$;

else

 Halve $\alpha^{(3)}$;

 Stop updating $\alpha^{(3)}$;

$\tilde{x}_i \leftarrow \tilde{x}_i + \alpha^{(3)} \vec{F}_i^{(3)}$;

until ($k^{(3)} > k_{\max}^{(3)}$) OR ($\|\vec{F}_i^{(3)}\| < \epsilon$);

Table 1

Common parameters used in simulations.

Deployed area	[0 m, 28 m] × [0 m, 28 m]
Total number of nodes N	196
Number of anchors	4
Distribution over area	Uniform

characteristic instead of simulating the whole network with a very large number of nodes. Each normal node has four reference anchors and has other normal nodes as reference neighbors. Table 1 defines the common parameters that we use in all four experiments. For convenience of demonstration, the values $\alpha^{(2)}$ and $\alpha^{(3)}$ are equal to $\alpha^{(1)}$. In this section, the normal nodes are updated in a random order to show the distributed ability of PPE.

4.1. Verifying the convergence of PPE

This simulation set is aimed at verifying the algorithm's robustness in both pre-refinement and refinement phases. We choose a normal node as the tested node and examine its movement trend under the force effect of the other normal nodes in the group after they undergo their raw estimations. The chosen nodes are tested in two situations: at the corner and in the middle of the deployed area of the group. The noise_{ij} in this subsection is Gaussian with $\sigma_n = 0.4$, the largest value in the chosen range.

Fig. 6(a) is the result of the first phase where each normal node uses anchors to produce its raw-estimation ($k_{\max}^{(1)} = 10$). The actual locations of all nodes are denoted as small dots while the errors are presented as solid lines. Fig. 6(b) illustrates the convergence image (or the representation of the movement tendency) of the corner tested node. Every position on the $[-7 \text{ m}, 35 \text{ m}] \times [-7 \text{ m}, 35 \text{ m}]$ grid with a grid step of 2 m (the deployed field is $[0 \text{ m}, 28 \text{ m}] \times [0 \text{ m}, 28 \text{ m}]$) is examined as if it is the current estimated position of the tested node. From each position, the tested node performs two successive position updates which are presented by two arrows to show its movement tendency. The figure shows that the tested node, at any current estimation and with its reference nodes suffering from a high degree of error, always moves to the unique balanced point which is close to its actual position.

Likewise, Fig. 6(c) is the estimation result of all normal nodes when phase 2 is finished with $k_{\max}^{(2)} = 10$ and Fig. 6(d) is the convergence image of the corner tested node with respect to the

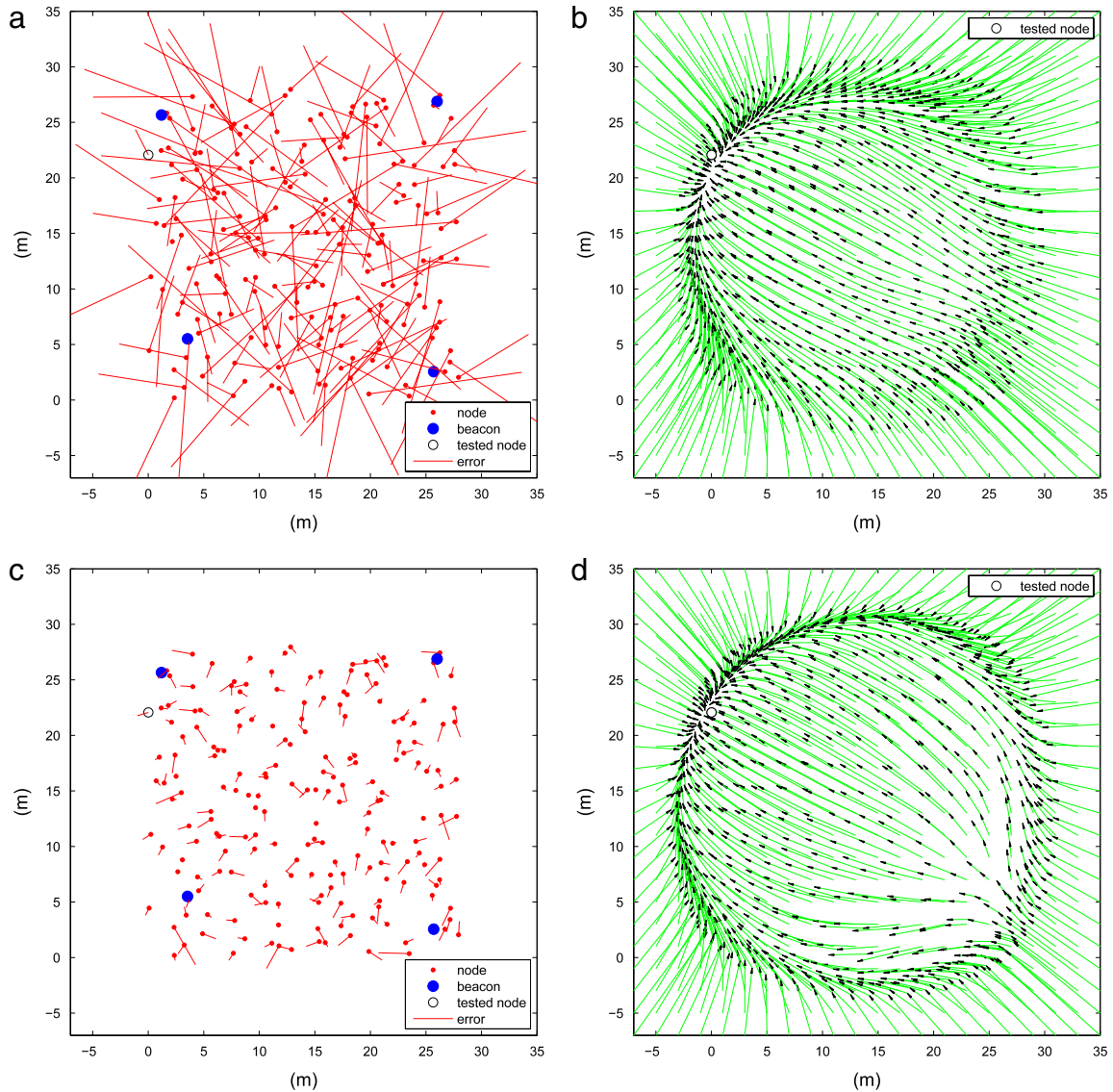


Fig. 6. (a) and (c) are the estimation results at the beginning and ending of phase 2, alternately, the errors of the estimated positions are denoted with solid lines. (b) and (d) are the movement tendency images of the corner tested node with respect to the estimation results in (a) and (c), alternately when Eq. (7) is used.

estimation result in Fig. 6(c). The errors of all normal nodes are reduced compared to the result of the raw estimation in Fig. 6(a). Comparing the two convergence images (Fig. 6(b) and (d)), the balanced positions of the corner tested node are almost the same. This confirms the fact that a node's updating is quite independent of those of others and this is the basis for the reduction in communication cost of PPE, as we mentioned in Section 3.4. This is also true when the tested node is in the middle of the deployed area, as depicted in Fig. 7. Fig. 7(a) and (b) are the convergence images of the middle tested node when phase 1 is complete and when phase 2 is complete, respectively. The movement tendency of a middle node is straighter and faster than that of a corner node. The balanced point is still unique and quite independent of the current estimation result of the reference neighbors.

The movement curve in the convergence images can be explained by the force effects of the small sub-areas of the reference nodes. Each of these areas attempts to keep away the tested node at a distance approximate to the distance from itself to the balanced point. Therefore, on the way to the balanced position, the tested node tends to move away from the side on which there are more reference nodes and creates the movement curve. Figs. 6 and 7 show that if the current estimated position is in the middle

of the group, then the convergence is fast and the movement is straight. This is why we use the average beacon position as the initial input position in phase 1.

Phase 3's convergence verification is demonstrated in Fig. 8, after phase 2 is complete. Fig. 8(a) is for the corner tested node, and Fig. 8(b) illustrates the middle tested node. It can be seen that if the current estimated position of the tested node has a large error, phase 3 still guarantees a correct convergence. The balanced position is unique and close to the actual position. Although the number of steps for the tested node to get to the balanced position is higher, the result of estimation is better. Obviously, all three phases of PPE guarantee network localization convergence even when the updates are ubiquitously performed on nodes. In other words, PPE is a robust distributed algorithm.

4.2. Necessity of the refinement phase when the distance error is proportional to the real range

In these simulations, we vary the standard deviation of noise e_j , σ_n , and use different distributions for noise e_j to show the improvement in the refinement phase over that in the pre-refinement phase. After the first ten iterations of phase 2, the

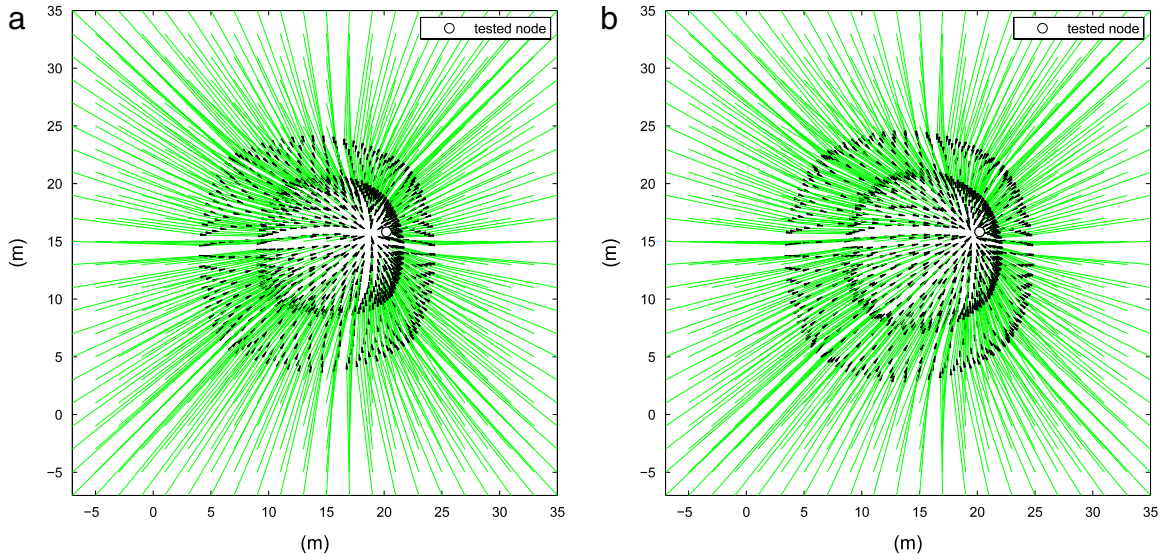


Fig. 7. (a) and (b) are the movement tendency images of the middle tested node with respect to the estimation results in Fig. 6(a) and (c), alternately when Eq. (7) is used.

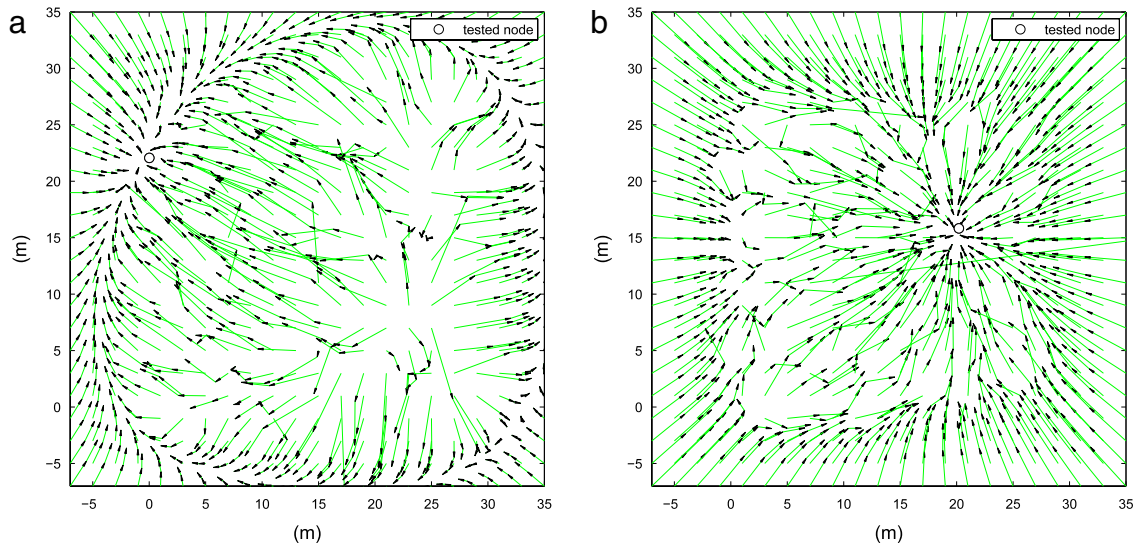


Fig. 8. (a) and (b) are the movement tendency images of the corner tested node and the middle tested node, alternately at the start of phase 3 in which Eq. (8) is used.

comparison between phase 2 and phase 3 occurs in the next 20 iterations. Apparently, the analysis in Section 3.3 is verified with the lower RMS error (see (12)) for phase 3 as shown in Fig. 9. Higher accuracy is gained with a very simple but efficient step which eliminates the distance factor from the force model.

Fig. 10(a) displays the average error result of 20 trials when phase 3 is used and not used after the 10th iteration. Gaussian, Rayleigh, and Uniform distributions for noise_{ij} are used, and 30 total iterations take place after the completion of phase 1. Although Uniform distributions for noise_{ij} are not appropriate in reality, we use it to emphasize that PPE does not depend on any specific distribution. Meanwhile, Fig. 10(b) plots the percentage of the improvement of phase 3 over that of phase 2. Generally, the error reduction is around 26%, a significant improvement. Fig. 10(a) does not only demonstrate the linear relation between the standard deviation σ_n of noise_{ij} and the RMS error of the solution, but also the independence of the result on the distribution of noise_{ij}. These consequences are consistent with (33), (37) and (38), in which the variance of the mean-force is linearly proportional to the variance of the individual force and does not depend on the distribution of the individual force. This independence is an important advantage

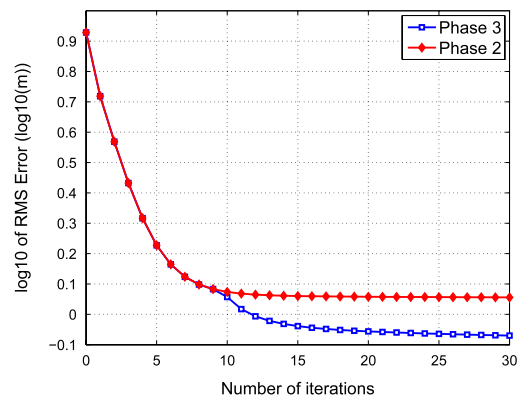


Fig. 9. RMS error of the estimated position over 30 iterations, the curve with diamonds represents the result of using the pre-refinement phase only while the curve with squares represents the result of refinement phase after ten iterations. Error is in log10 of meters.

point of PPE whereas other successive refinement methods like MLE [6,20] or MDS variations [2,1,4] depend strongly on the noise

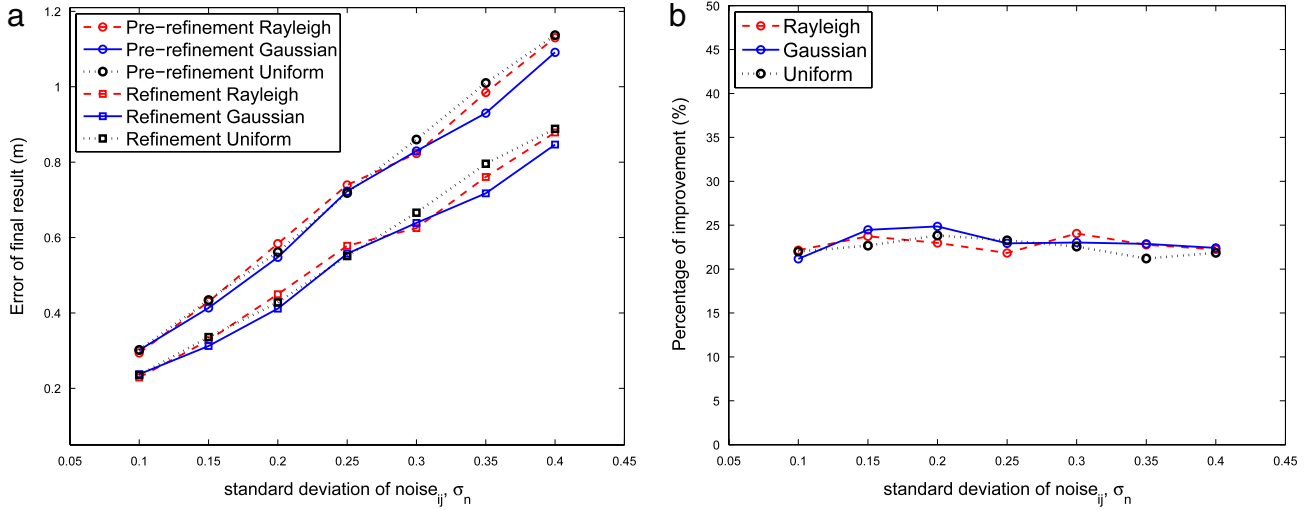


Fig. 10. Error and accuracy improvement over σ_n , (a) Result error with and without refinement phase. (b) Percentage of improvement when the refinement phase is used.

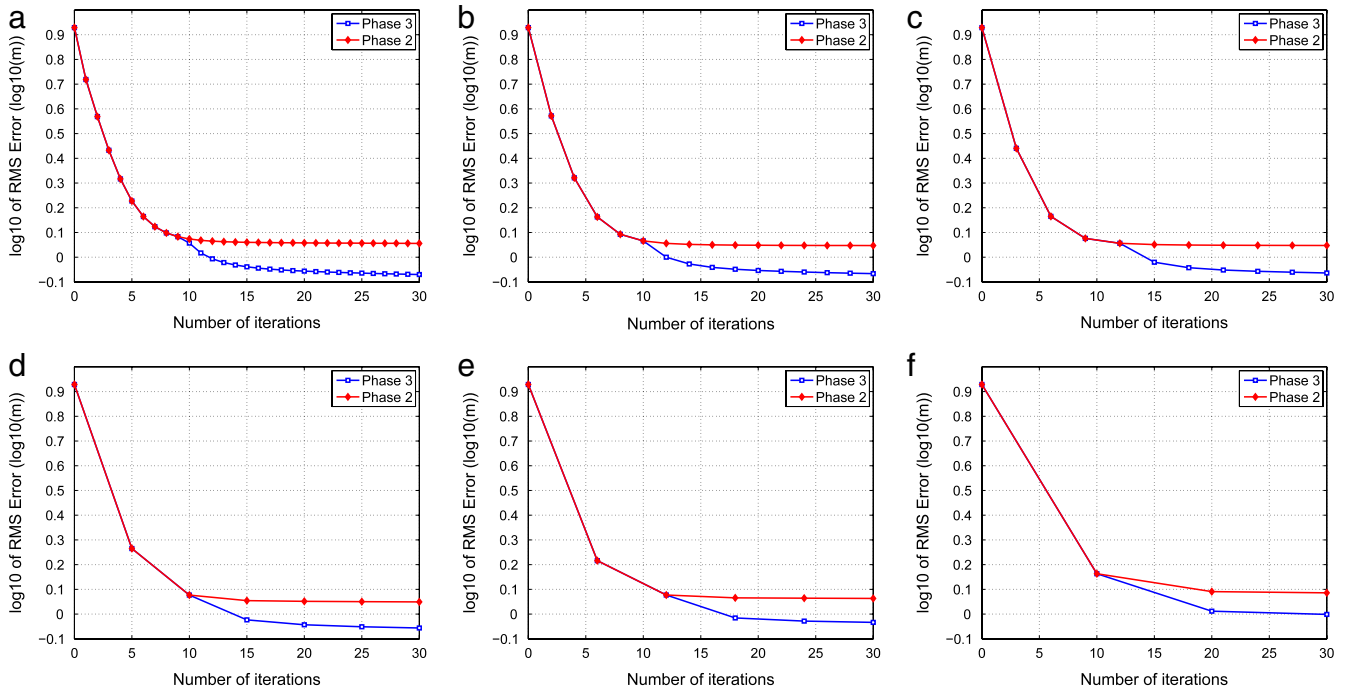


Fig. 11. Illustration of communication cost reduction when the cycle request number $R = 1, 2, 3, 5, 6,$ and $10,$ alternately.

model. For these approaches, the formula of the measurement error distribution must be known in order to derive the formula of the result error and then to find the calculation to force this error to a minimum.

4.3. Communication reduction

This experiment is performed to examine the ability of PPE to reduce communication cost. Each sub-figure in Fig. 11 is the average result of 20 trials, with the total iteration number of each trial being 30 (iterations of phase 1 are not included). The sub-figures correspond to the different cycle request values R : 1, 2, 3, 5, 6, and 10. Fig. 11 again shows that the convergence of PPE is quite independent of the current position at the reference

nodes. Apparently, achieving high accuracy incurs a trade-off of the number of iterations and the communication cost. Nonetheless, it is worth reducing the communication cost, the most critical cost, especially when the trade-off error increases insignificantly as in Fig. 12. Increasing the cycle request number R is very efficient when this number is not very high (2, 3 or 5). The RMS error then increases insignificantly (0.94%, 1.52% or 3.2%) while the communication cost decreases remarkably (50%, 75% or 80%) compared to the error when a data exchange occurs with every node position update ($R = 1$). Even when the number of internal iterations between two successive data exchanges is 9 ($R = 10$), the error would increase only 17.1%. Hence, based on the specific requirement of designed networks in terms of communication cost and accuracy, an appropriate R can be chosen.

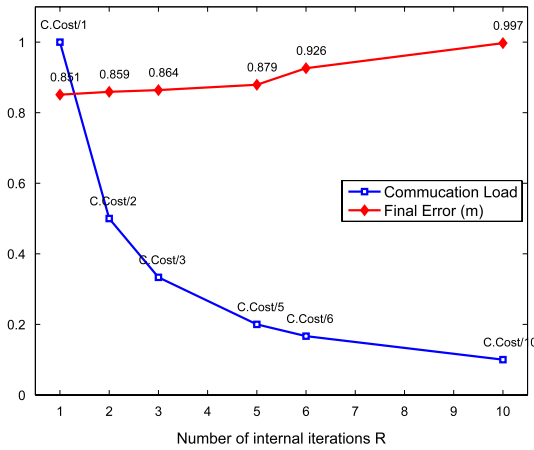


Fig. 12. Cost communication and final RMS error over the number of iterations. “C.Cost” is the maximum communication cost when the cycle request number R is 1.

4.4. Comparison to MLE and dwMDS

The contemporary popular MLE and dwMDS methods aim to minimize the stress function [2,6]

$$S = \sum_{i=1}^{N-1} \sum_{j=i+1}^N w_{ij}(\tilde{d}_{ij} - \delta_{ij})^2, \tag{48}$$

where w_{ij} is the weight value.

For the dwMDS (distributed weighted Multi Dimensional Scaling) simulation, we rely mainly on Costa’s paper [2] and a part of the Matlab code available on Patwari’s website [12] to solve our problems and compare the results with our PPE’s. For MLE (Maximum Likelihood Estimation), the updating method is the nonlinear conjugate gradient. dwMDS is actually a two-step method in which the first step is raw estimation, and the second is refined estimation. However, the convergence time of this algorithm is large; the result of the first step suffers from negative estimation (the estimated positions make the deployed area look shrunk); and the accuracy is poor especially when the measurement error has the log-normal model. Meanwhile, MLE’s estimation would hardly converge with random initial positions of the normal nodes. Therefore, we take the output of PPE’s phase 1 as the raw estimation inputs for both MLE and dwMDS although this unbiased estimation is one of the important advantages of PPE over other algorithms. Especially for the case of log-normal distance error, we use the actual positions of all nodes as the input for the second step of dwMDS so that the simulation time is smaller and the accuracy is higher. Even so, the simulation results illustrate that the performance of PPE is superior to those of the other methods.

4.4.1. Measurement error proportional to the real distance case

The measurements are generated according to the noise model in (2) where $noise_{ij}$ is Gaussian for comparison purpose. σ_n varies from 0.1 to 0.4 in intervals of 0.05, as in Fig. 13. We use the same set of actual node positions, and with each value of σ_n , we create 20 sets of measurement data based on the error distribution. For dwMDS and MLE, the algorithms stop if the maximum iteration number reaches 80 or if the average updating change of the stress function is less than 10^{-4} . The stop conditions for PPE are $k_{max}^{(1)} = 10$, $k_{max}^{(2)} = 10$, $k_{max}^{(3)} = 30$, or the movement distance is less than 10^{-4} m. Fig. 13 shows that PPE gives the lowest RMS error at every value of σ_n . This means that PPE has a better performance even when dwMDS and MLE use the unbiased raw estimation of PPE’s phase 1. Moreover, PPE also attempts to eliminate d_{ij} from the

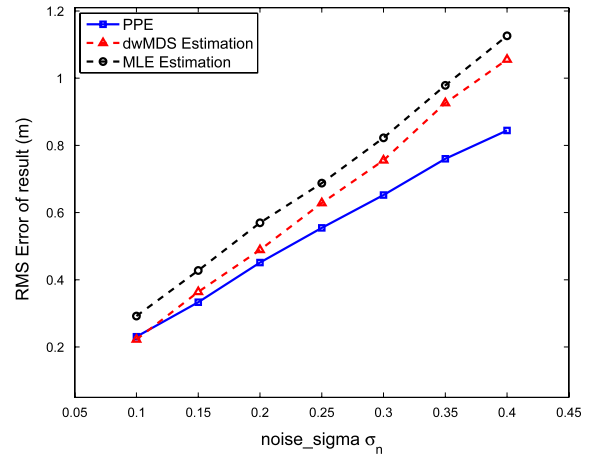


Fig. 13. RMS errors of different methods over the standard deviation σ_n of $noise_{ij}$, for the error model in Eq. (2).

Table 2

Computation cost.		
	Multiplications per iteration	Iteration number
dwMDS	$3(M_i^{(3)})^2 + 13M_i^{(3)}$	80
MLE	$7M_i^{(3)} + 3M_i^{(3)} * Iter_0$	52
PPE	$5M_i^{(3)}$	40

error, turning the variance of $d_{ij}, noise_{ij}$ into the smaller variance of $\tilde{d}_{ij}, noise_{ij}$ (see (39)). As a consequence, the accuracy of PPE in the range-based scheme whose error of measurement is modeled in (2) is better.

Table 2 shows the approximated computation costs of each normal node for three methods in the simulation. The values in the second column are the approximated numbers of multiplications in one iteration. We consider a square-root operation as a multiplication and ignore several extra multiplications outside the loops. The values in the third column are the average numbers of iterations necessary to produce the final result when the input data is the raw estimation. $Iter_0$ is the extra number of iterations used in MLE for computing the line search optimization. The average of this number or $mean(Iter_0)$ is 5.6. Table 2 also shows that PPE requires fewer iterations than do dwMDS and MLE to reach a better result. In other words, PPE needs less of both communication and computation costs even when the cycle request number in this simulation is 1.

4.4.2. Log-normal distribution for the measurement error case

For this part, sets of measurements modeled in (47) are used instead of those in (2). σ_p is the standard deviation of $Pnoise$ and it is varied from 1 to 4 in increment of 0.5. We repeat that with log-normal error model, only two first phases of PPE with the de-biasing function $\log_{10}(\cdot)$ are used. MLE has the same stop condition as in the previous experiment while PPE stops at $k_{max}^{(1)} = 20$, $k_{max}^{(2)} = 40$, or when the movement distance is less than 10^{-4} m. It should be noted that the input data for PPE does not include the information of normal nodes’ locations, while dwMDS uses the real node positions as the raw input and MLE uses PPE’s raw estimation provided by phase 1. Therefore, for dwMDS, we set the maximum number of iterations to be 20 in this case because a small value of this number leads to small final error. However, it can be seen in Fig. 14 that dwMDS gives the worst error compared to those of PPE and MLE which have equivalently good results at small σ_p . Although MLE has the best result when $\sigma_p = 1$, MLE’s accuracy decreases faster than that of PPE while PPE’s error maintains a

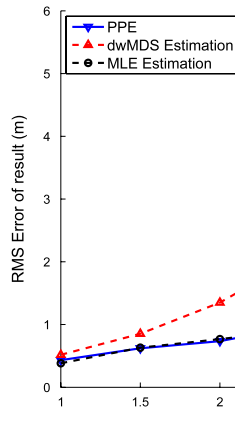


Fig. 14. RMS errors of different methods over the standard deviation σ_p of $Pnoise$, for the error model in Eq. (47).

linear relation with σ_p when σ_p is large ($\sigma_p > 3$). Obviously, MLE is more sensitive to the big log-normal noise. Even with the same raw estimation, and when the stop conditions are tougher for high accuracy, MLE is not as good as PPE when σ_p increases. It is known that large values of σ_p are the case of the practical implementation, especially for indoor settings.

5. Conclusions

Localization problems in WSNs include numerous challenges for network designers in terms of accuracy, low-cost computation, low-cost communication, distributed ability, scalability, etc. In this article, we introduced a new method, the Push-pull Estimation, which has many advantages that localization solvers aim at. Each normal node, with the range measurements to its reference nodes and the current estimated distances, models the differences to forces and then moves under the effect of these forces. The algorithm which relies mainly on geometry is not based on available mathematical methods, such as MLE, MDS, Particle Filter, etc. As a result, we have to pay more attention to analyzing and proving most of PPE's characteristics as well as its related problems. Simulation experiments do not only confirm the above analysis, but also show that PPE is a robust distributed algorithm with very simple calculations and a low communication cost. This cost can even be reduced considerably with an insignificant accuracy trade-off by increasing the cycle request number. Furthermore, the method has a flexibility to deal with different unbiased distributions of measurement error, especially when the error depends on the actual distance or when the error has a log-normal distribution. If the measurement error is unbiased and is proportional to the actual range, PPE's result is better than that of the popular successive refinement representatives MLE and dwMDS. With the log-normal distribution of the measurement interfered by noise, PPE is far better than dwMDS in terms of computational cost and accuracy. It is also more robust than MLE, particularly if the power error has large variance, even when MLE uses its raw estimation as the input. Moreover, PPE needs neither complex calculations for these initial data nor final corrections like the re-scaling Procrustes algorithm to improve result accuracy, thus the calculation and communication are reduced and simpler. Therefore, PPE is a very good localization candidate to WSN designers.

Acknowledgments

This research was supported by the MKE (Ministry of Knowledge and Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the

IITA (Institute of Information Technology Advancement) (IITA-2010-(C1090-1002-0003)) and by Basic Science Research Program Through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0076798).

Appendix

Main parameters

i, j, k : nodes' names, also used as the index related to the nodes.
 i_0, k_0 : actual position of node i , node k .

N, n, m : total number of nodes, number of normal nodes and number of anchors respectively, $N = n + m$.

x_i : true position of node i , $x_i \in \mathbf{R}^2$.

\tilde{x}_i : current estimated position of node i , $\tilde{x}_i \in \mathbf{R}^2$.

d_{ij} : actual distance from x_i to x_j .

\tilde{d}_{ij} : current calculated distance from \tilde{x}_i to \tilde{x}_j .

δ_{ij} : measured distance, or measurement from node i to node j which is estimated by the network and serves as the input for PPE algorithm.

p : index of PPE's phases; $p = 1, 2$ or 3 .

$\vec{f}_{ij}^{(p)}$: individual force caused by j on i . $\vec{f}_{ij}^{(1)}$, $\vec{f}_{ij}^{(2)}$, and $\vec{f}_{ij}^{(3)}$ are defined in Eqs. (6), (7) and (8) respectively.

\vec{e}_{ij} : unit vector pointing the direction from \tilde{x}_i to \tilde{x}_j .

$\vec{e}_{ij} = \frac{1}{d_{ij}}(\tilde{x}_j - \tilde{x}_i)$.

$\vec{F}_i^{(p)}$: the mean-force, defined in (4).

$M_i^{(p)}$: number of node i 's related nodes. $M_i^{(1)}$ is number of i 's related beacons, $M_i^{(2)}$ is number of both i 's related beacons and i 's neighbors, and $M_i^{(3)} = M_i^{(2)}$.

$\alpha^{(p)}$: movement rate.

L_{ij} : variable defined in (10) to unify the form of $\vec{f}_{ij}^{(p)}$ for statement (iii)'s proof.

J, K : center points of sub-areas.

$(u_i, v_i), (u_j, v_j)$: positions of node i and node j in the coordinate system uOv .

$(\Delta F_u, \Delta F_v)$: coordinate of $\vec{\Delta F}_i$ in the coordinate system uOv .

\vec{F}_i : variable replacing $\vec{F}_i^{(p)}$ in statement (i)'s proof.

P_{ij}, P_{noise} : received power (dB) and error power (dB) at i respectively when j transmits.

$g(\cdot)$: de-biasing function.

$k^{(p)}, k_{max}^{(p)}$: iteration counting variable and its maximum value at phase p .

References

- [1] S. Biaz, Y. Ji, Precise distributed localization algorithms for wireless networks, in: WOWMOM'05: Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM'05, IEEE Computer Society, Washington, DC, USA, 2005.
- [2] J.A. Costa, N. Patwari, A.O. Hero III, Distributed weighted multidimensional scaling for node localization in sensor networks, ACM Transactions on Sensor Networks 2 (2006) 39–64.
- [3] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking MobiCom'03, 2003.
- [4] G. Latsoudas, N.D. Sidiropoulos, A fast and effective multidimensional scaling approach for node localization in wireless sensor networks, IEEE Transactions on Signal Processing 55 (2007) 5121–5127.
- [5] V.-D. Le, V.-H. Dang, S. Lee, S.-H. Lee, Distributed localization in wireless sensor networks based on force-vectors, in: ISSNIP 2008: Intelligent Sensors, Sensor Networks and Information Processing, 2008.

- [6] X. Li, Collaborative localization with received-signal strength in wireless sensor networks, *IEEE Transactions on Vehicular Technology* 56 (2007) 3018–3022.
- [7] R.A. Malaney, Nuisance parameters and location accuracy in log-normal fading models, *IEEE Transactions on Wireless Communications* 6 (2007) 937–947.
- [8] J.F. Michel, M. Vossiek, Wireless sensor network approach for robust localization of mobile nodes with minimal complexity, *e & i Elektrotechnik und Informationstechnik* 125 (2008) 341–346.
- [9] O.L. Moses, D. Krishnamurthy, R. Patterson, A self-localization method for wireless sensor networks, *EURASIP Journal on Applied Signal Processing* 4 (2003) 348–358.
- [10] S. Namik, U.J. Ferner, K.W. Sowerby, Localization in harsh propagation environments, in: *Communications Theory Workshop, 2008, AusCTW 2008, Australian, 2008*, pp. 161–166.
- [11] D. Niculescu, B. Nath, Dv based positioning in ad hoc networks, *Journal of Telecommunication Systems* 22 (2003) 267–280.
- [12] N. Patwari, Wireless sensor network localization measurement repository. <http://www.eecs.umich.edu/hero/localize/>.
- [13] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, *Signal Processing Magazine, IEEE* 22 (2005) 54–69.
- [14] N. Patwari, A.O. Hero III, Adaptive neighborhoods for manifold learning based sensor localization, in: *2005 IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, 2005*, pp. 1098–1102.
- [15] N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, R.J. Odea, Relative location estimation in wireless sensor networks, *IEEE Transactions on Signal Processing* 51 (2003) 2137–2148.
- [16] H. Qasem, L. Reindl, Precise wireless indoor localization with trilateration based on microwave backscatter, in: *Wireless and Microwave Technology Conference, 2006. WAMICON'06. IEEE Annual, 2006*, pp. 265–270.
- [17] A. Savvides, H. Park, M.B. Srivastava, The bits and flops of the n -hop multilateration primitive for node localization problems, in: *WSNA'02: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ACM Press, 2002, pp. 112–121.
- [18] J.-P. Sheu, P.-C. Chen, C.-S. Hsu, A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement, *IEEE Transactions on Mobile Computing* 7 (2008) 1110–1123.
- [19] F. Sivrikaya, B. Yener, Time synchronization in sensor networks: a survey, *IEEE Network* 18 (2004) 45–50.
- [20] J. Tabrikian, J.L. Krolik, H. Messer, Robust maximum likelihood source localization in an uncertain shallow water waveguide, *Journal of the Acoustical Society of America* 101 (1997) 241–249.
- [21] P. Tarrío, A.M. Bernardos, J.R. Casar, An RSS localization method based on parametric channel models, in: *International Conference on Sensor Technologies and Applications, 2007*, pp. 265–270.
- [22] S. Tian, S. Zhang, X. Wang, P. Sun, H. Zhang, A selective anchor node localization algorithm for wireless sensor networks, in: *ICIT'07: International Conference on Convergence Information Technology, IEEE, 2007*, pp. 358–362.
- [23] C. Wang, L. Xiao, Sensor localization under limited measurement capabilities, *IEEE Network* (2008) 16–23.
- [24] H.-W. Wei, Q. Wan, Z.-X. Chen, S.-F. Ye, A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location, *IEEE Transactions on Signal Processing* 56 (2008) 3018–3022.
- [25] J. Yang, Y. Chen, A theoretical analysis of wireless localization using RF-based fingerprint matching, in: *Proceedings of the 22nd IEEE International Parallel & Distributed Processing Symposium, IPDPS, 2008*.

- [26] S. Zainalie, A clustering algorithm for localization in wireless sensor networks, in: *2008 International Symposium on Telecommunications, 2008*, pp. 435–439.



Viet-Hung Dang received his B.S. from HoChiMinh University of Technology in 2003, HoChiMinh City, Vietnam. He got his M.S. from HoChiMinh University of Technology 2 years later. He is currently working towards a Ph.D. degree in Computer Science at Kyung Hee University, Korea since 2008. His research interests are Distributed computing, Machine learning and Localization.



Viet-Duc Le received his B.S. from HoChiMinh University of Technology in 2002, HoChiMinh City, Vietnam. He received his M.S. degree in Computer Science at Kyung Hee University, Korea in 2009. His research interests are Localization, Electronic control and Machine learning.



Young-Koo Lee received his B.S., M.S., and Ph.D. degrees in Computer Science from the Korea Advanced Institute of Science and Technology, Korea. He is a professor in the Department of Computer Engineering at Kyung Hee University, Korea. His research interests include ubiquitous data management, data mining, and databases. He is a member of IEEE, the IEEE Computer Society, and the ACM.



Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Dept. of Computer Engineering, Kyung Hee University, Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been affiliated with a director of Neo Medical ubiquitous- Life Care Information Technology Research Center, Kyung Hee University since 2006. Before joining Kyung Hee University, he was an assistant professor in the Dept. of Comp. Sci., Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on Ubiquitous Computing and applications, Context-aware Middleware, Sensor Operating Systems, Real-Time Systems and Embedded Systems. He is a member of the ACM and IEEE.